Network Coding Capacity Regions via Entropy Functions

Terence H. Chan and Alex Grant

Abstract

In this paper, we use entropy functions to characterise the set of rate-capacity tuples achievable with either zero decoding error, or vanishing decoding error, for general network coding problems. We show that when sources are colocated, the outer bound obtained by Yeung, *A First Course in Information Theory*, Section 15.5 (2002) is tight and the sets of zero-error achievable and vanishing-error achievable rate-capacity tuples are the same. We also characterise the set of zero-error and vanishing-error achievable rate capacity tuples for network coding problems subject to linear encoding constraints, routing constraints (where some or all nodes can only perform routing) and secrecy constraints. Finally, we show that even for apparently simple networks, design of optimal codes may be difficult. In particular, we prove that for the incremental multicast problem and for the single-source secure network coding problem, characterisation of the achievable set is very hard and linear network codes may not be optimal.

I. Introduction

Determining network coding capacity regions (the set of link capacities and source rates admitting a network coding solution for a given multicast) is a fundamental problem in information theory and communications. Recently, the network coding capacity region for general network coding problems was implicitly determined using entropy functions [1][1]. This characterisation has a similar (but slightly more complicated) form to the outer bound in [2, Section 15.5], which is expressed in terms of almost entropic functions. Explicit characterisation of the set of entropy functions (or its closure) is however a very difficult open problem (for instance, it is known that this set is not polyhedral [3]).

It is therefore natural to wonder whether there might be a simpler, explicit characterisation of network coding capacity regions which somehow avoid the use of entropy functions. However, these two problems are inextricably linked, and in general, determining network coding capacity regions is as hard as finding the set of all entropy functions, or equivalently, determining all information inequalities [4].

One approach to avoid these intrinsic challenges of the general case is to seek special cases, or specific classes of networks for which an explicit, computable solution is possible. To date, only a few such special cases have been found. One notable example is where a single source data stream is unicast to multiple destinations. In this case, the capacity region is characterised by graph-theoretic maximal flow/minimum cut bounds, and linear codes

[1]We assume the reader is familiar with polymatroids, entropy functions and representable functions. We review these and other related concepts in Section II and as needed throughout the paper.

are optimal [5]. As second example is a secure network coding problem when all links have equal capacities and the eavesdropper's capability is only limited by the total number of links it can wiretap. In this case, the minimum cut bound is also tight [6].

Another approach is to develop computable bounds on the capacity region. Relaxation from entropy functions to polymatroids yields the so-called linear programming bound [2, Section 15.6]. Although this bound is explicit, both the number of variables and the number of constraints increase exponentially with the number of links in the network, making the bound computationally infeasible even for modest networks. Other works such as [7], [8], [9] aim to obtain useful outer bounds with computationally efficient algorithms for their evaluation. In fact, it can be shown that all of the bounds obtained in those works are relaxations of the linear programming bound from [2, Section 15.6].

This paper extends [1] in several aspects. In both [1] and [2], vanishing decoder error probabilities are allowed. In Section III, we will extend these results to the case where the decoding error probability must be exactly zero. We also prove that when all the sources are colocated, the outer bound [2] is in fact zero-error achievable and tight. For the general non-colocated source case, we show that tightness of the outer bound reduces to a question of whether or not the addition of a zero-rate link can change the capacity region of a particular network that we derive from the original network. This leads us to conjecture tightness of the bound in general.

The existing capacity result [1] does not place any constraints on the operation of intermediate nodes, allowing arbitrary network coding operations. We further extend this entropy-function based approach to three different practically-motivated cases where additional constraints are placed on the network: (a) all nodes use linear encoding, (b) all, or some nodes can only perform routing, and (c) we desire secrecy in the presence of an eavesdropper.

In Section IV, we consider the case where only linear network codes are allowed. We prove equivalence of zero-error and vanishing-error achievability, and characterise the linear network coding capacity region using representable functions.

When comparing the performance of network coding to routing-only networks (where nodes can only store and forward received packets), it may be useful to have a capacity characterisation for routing in terms of entropy functions. In Section V we introduce almost atomic functions which provide just such a characterisation. We go on to consider heterogeneous networks, containing both network coding nodes and routing nodes, and show how to obtain an entropy function characterisation of the capacity region.

In Section VI, we impose secrecy constraints, where we assume the presence of eavesdropper who has access to certain links and desires to decode particular sets of sources. The objective is to design a transmission scheme such that the eavesdropper remains ignorant of the source messages. We will once again characterise the resulting general secure network coding capacity region via representable functions.

Finally, in Section VII, we will consider two very simple network coding problems where despite the apparent simplicity of the setup, characterisation of the capacity region turns out to be extremely difficult, and linear codes may be suboptimal. The first is incremental multicast, where the sources and sinks are ordered such that sink i demands sources $1, 2, \ldots, i$. The second example is secure unicast of a single source. This demonstrates that the

seemingly innocuous addition of a security constraint loosens the minimum cut bound [5]. Similarly we see that the min-cut result from [6] does not hold even for this simple case.

Notation: $\mathbb{R}$ is the set of all real numbers and $\mathbb{R}_0$ is the set of all nonnegative real numbers. Random variables will be denoted by uppercase roman letters X and sets will be denoted using uppercase script $\mathcal{X}$. The power set $2^{\mathcal{X}}$ is the set of all subsets of $\mathcal{X}$. For a discrete random variable X taking values in the set (or alphabet) $\mathcal{X}$, its *support* $\mathsf{SP}(X)$ is

$$\mathsf{SP}(X) \triangleq \{x \in \mathcal{X} : \Pr(X = x) > 0\}.$$

Realisations of a random variable will typically denoted via lowercase x.

For sets $\{X_1, X_2, \dots, X_n\}$ and $\mathcal{S} \subseteq \{1, 2, \dots, n\}$, the subscript notation $X_{\mathcal{S}}$ will mean $\{X_i, i \in \mathcal{S}\}$. Where it will cause no confusion, set notation braces will be omitted from singletons and union will be denoted by juxtaposition. Thus $\mathcal{A} \cup \mathcal{B} \cup \{i\}$ can be written $\mathcal{A}\mathcal{B}i$ and so on. Ordered tuples will be denoted

$$(x(i), i = 1, 2, \dots, n) = (x_1, x_2, \dots, x_n).$$

II. Background

In this section we provide the formal problem definition for transmission of information in networks consisting of error-free broadcast links. This includes representation of such networks as hypergraphs, the notions of a multicast connection requirement, network codes and zero-error or vanishing-error achievability. We then review existing results on characterisation of the network coding capacity via the use of entropy functions. In this section, we do not impose any additional constraints (such as linearity or security) beyond zero- or vanishing-decoding-error probability.

A. Unconstrained Network Coding for Broadcast Networks

We represent a communication network by a directed hypergraph $\mathsf{G} = (\mathcal{V}, \mathcal{E})$. The set of nodes

$$\mathcal{V} = \left\{V_1, \dots, V_{|\mathcal{V}|}\right\}$$

and the set of hyperedges

$$\mathcal{E} = \left\{E_1, \dots, E_{|\mathcal{E}|}\right\}$$

respectively model the set of communication nodes and error-free broadcast links. In particular, each hyperedge $e \in \mathcal{E}$ is defined by a pair $(\mathsf{tail}(e), \mathsf{head}(e))$, where $\mathsf{tail}(e) \in \mathcal{V}$ is the transmit node and $\mathsf{head}(e) \subseteq \mathcal{V}$ is the set of nodes which receive identical error-free transmissions from $\mathsf{tail}(e)$. When $\mathsf{head}(e)$ is a singleton, e models an ordinary point-to-point link.

We assume that the network is free of directed cycles (a nonempty sequence of links $\{f_1, \dots, f_k\}$ such that $\mathsf{tail}(f_i) \in \mathsf{head}(f_{i-1})$ for $i = 2, \dots, k$ and $\mathsf{tail}(f_1) \in \mathsf{head}(f_k)$).

Definition 1 (Connection Constraint): For a given communication network G, a *connection constraint* M is a tuple $(\mathcal{S}, O, D)$, where $\mathcal{S}$ indexes the sources, $O : \mathcal{S} \mapsto 2^{\mathcal{V}}$ specifies the source locations and $D : \mathcal{S} \mapsto 2^{\mathcal{V}}$ specifies the sink nodes. Unless specified otherwise, we let

$$\mathcal{S} = \{S_1, \ldots, S_{|\mathcal{S}|}\}$$

be the index set of $|\mathcal{S}|$ independent sources. Source $s \in \mathcal{S}$ is available at every node in $O(s) \subseteq \mathcal{V}$. Note that in general each source can be available to more than one network node. The sink nodes $D(s) \subseteq \mathcal{V}$ are nodes where source s should be reconstructed according to some desired error criteria.

It is conceptually useful to imagine each source s as a message sent along an imaginary source edge, which for simplicity will also be labelled s. In this case, we can use the notation $\mathsf{head}(s)$ to denote $O(s)$. For any $e \in \mathcal{E}$ and $u \in \mathcal{V}$, we define

$$\mathsf{in}(e) \triangleq \{f \in \mathcal{S} \cup \mathcal{E} : \mathsf{tail}(e) \in \mathsf{head}(f)\} \tag{1}$$

$$\mathsf{in}(u) \triangleq \{f \in \mathcal{S} \cup \mathcal{E} : u \in \mathsf{head}(f)\}. \tag{2}$$

In other words, $\mathsf{in}(\cdot)$ is the set of incoming edges (including the imaginary source edges).

Solution of the network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$ requires a transmission scheme allowing source s to be reliably reconstructed at the sink nodes $D(s)$. We have not yet specified the transmission capacity of each hyperedge, or the rate of each source. Characterisation of the capacity region for network coding means determination of the combinations of source rates and hyperedge capacities which admit a network coding solution. Before we can proceed, we need to formalise what we mean by *network code*, *reliable* and *rate-capacity tuple*.

Definition 2 (Network Code): A network code

$$\Phi \triangleq \{\phi_e : e \in \mathcal{E}\} \tag{3}$$

for the problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$ is a set of local encoding functions

$$\phi_e : \prod_{f \in \mathsf{in}(e)} \mathcal{Y}_f \mapsto \mathcal{Y}_e.$$

where $\mathcal{Y}_s$ is the alphabet of source $s \in \mathcal{S}$ and $\mathcal{Y}_e$ is the alphabet for messages transmitted on hyperedge $e \in \mathcal{E}$. Each network code induces a set of random variables

$$\{Y_f, f \in \mathcal{S} \cup \mathcal{E}\}. \tag{4}$$

as follows:

1) $\{Y_s, s \in \mathcal{S}\}$ is a set of mutually independent random variables, each of which is uniformly distributed over its support and denotes a message generated by a source.
2) For each $e \in \mathcal{E}$,

$$Y_e = \phi_e(Y_f : f \in \mathsf{in}(e)) \tag{5}$$

and denotes the message transmitted on hyperedge $e \in \mathcal{E}$.

If the set of random variables induced by a network code is given, then the local encoding functions (3) are determined with probability one. In other words, if

$$\Pr\left(Y_f = y_f, f \in \mathsf{in}(e)\right) > 0,$$

then for all $y_e \neq \phi_e\left(y_f : f \in \mathsf{in}(e)\right)$,

$$\Pr\left(Y_e = y_e \mid Y_f = y_f, f \in \mathsf{in}(e)\right) = 0.$$

For this reason, we will often specify a network code by its set of induced random variables.

The following lemma follows directly from the above definitions and gives a necessary and sufficient condition under which a set of random variables is induced by a network code.

Lemma 2.1: A set of random variables $\{Y_f, f \in \mathcal{S} \cup \mathcal{E}\}$ defines a network code, with respect to a network coding problem P, if and only if

1) Y_s is uniformly distributed over its support for all $s \in \mathcal{S}$.
2) $H\left(Y_s, s \in \mathcal{S}\right) = \sum_{s \in \mathcal{S}} H\left(Y_s\right)$.
3) $H\left(Y_e \mid Y_f : f \in \mathsf{in}(e)\right) = 0$ for all $e \in \mathcal{E}$.

Conditions 2) and 3) are due to the mutual independence of the sources and the deterministic encoding constraints.

Definition 3 (Rate-Capacity Tuples): For a network coding problem P let

$$\chi(\mathsf{P}) \triangleq \mathbb{R}_0^{|\mathcal{S}|} \times \mathbb{R}_0^{|\mathcal{E}|}.$$

be the set of all *rate-capacity tuples*

$$(\lambda, \omega) = (\lambda(s) : \ s \in \mathcal{S}, \ \omega(e) : \ e \in \mathcal{E})$$

for P.

Definition 4 (Fitness): A rate-capacity tuple $(\lambda, \omega) \in \chi(\mathsf{P})$ is *fit* for a network code $\{Y_f, f \in \mathcal{S} \cup \mathcal{E}\}$ on P if there exists $c > 0$ such that for all $e \in \mathcal{E}$ and $s \in \mathcal{S}$,

$$\lambda(s) \leq c \log |\mathsf{SP}(Y_s)|, \tag{6}$$

$$\omega(e) \geq c \log |\mathsf{SP}(Y_e)|. \tag{7}$$

The tuple is *asymptotically fit* for a sequence of network codes $\{Y_f^n, f \in \mathcal{S} \cup \mathcal{E}\}$ for $n = 0, 1, \ldots$ if there exists a sequence $c_n > 0$ such that for all $e \in \mathcal{E}$ and $s \in \mathcal{S}$,

$$\lim_{n \to \infty} c_n \log |\mathsf{SP}(Y_s^n)| \geq \lambda(s), \tag{8}$$

$$\lim_{n \to \infty} c_n \log |\mathsf{SP}(Y_e^n)| \leq \omega(e). \tag{9}$$

Note that fitness does not imply achievability of a rate-capacity tuple, rather that the tuple is not impossible. Fitness indicates that (up no normalisation) each individual source rate is not too large to be achieved by the corresponding source variable with the given alphabet size, and that each hyperedge capacity is large enough to carry the corresponding edge variable regardless of particular distribution.

Definition 5 (Zero-error Achievable Rate-Capacity Tuples): A rate capacity tuple

$$(\lambda, \omega) = (\lambda(s) : \ s \in \mathcal{S}, \ \omega(e) : \ e \in \mathcal{E})$$

is called *zero-error achievable*, or 0-*achievable* if there exists a sequence of network codes $\Phi^n, n = 1, 2, \ldots$ and corresponding induced random variables $\{Y_f^n : f \ \in \mathcal{E} \cup \mathcal{S}\}$ such that

1) (λ, ω) is asymptotically fit for Φ^n.
2) for any source $s \in \mathcal{S}$ and receiver node $u \in D(s)$, the source message Y_s^n can be uniquely determined from the received messages $\left(Y_f^n : f \in \mathsf{in}(u)\right)$. In other words,

$$H\left(Y_s^n \mid Y_f^n, f \in \mathsf{in}(u)\right) = 0. \tag{10}$$

In Definition 5, each network code in the sequence has zero probability of decoding error. Relaxing this criteria to allow decoding error probability that vanishes in the limit, we have the following definition.

Definition 6 (Vanishing Error Achievable): A rate capacity tuple (λ, ω) is called *vanishing error achievable*, or ϵ-achievable if the tuple is asymptotically fit, and

$2'$) for all $s \in \mathcal{S}$ and $u \in D(s)$, there exists decoding functions $g_{s,u}^n$ such that

$$\lim_{n \to \infty} \Pr(Y_s^n \neq g_{s,u}^n(Y_f^n : f \in \mathsf{in}(u))) = 0.$$

In other words, decoding error probabilities vanish asymptotically.

For any subset $\mathcal{R} \subseteq \chi(\mathsf{P})$, define $\mathsf{CL}(\mathcal{R})$ as the subset of $\chi(\mathsf{P})$ containing all tuples (λ, ω) such that there exists a sequence of $(\lambda^n, \omega^n) \in \mathcal{R}$ and positive numbers c_n satisfying

$$\lim_{n \to \infty} c_n \omega^n(e) \leq \omega(e), \tag{11}$$

$$\lim_{n \to \infty} c_n \lambda^n(s) \geq \lambda(s). \tag{12}$$

Clearly, if every tuple in $\mathcal{R}$ is 0-achievable/ϵ-achievable, then $\mathsf{CL}(\mathcal{R})$ is also 0-achievable/ϵ-achievable.

The central theme of this paper is the characterisation of 0-achievable and ϵ-achievable regions for network coding via the use of *entropy functions*.

Definition 7 (Entropy Function): A set of random variables $\{Y_i, i \in \mathcal{N}\}$ (where $\mathcal{N}$ is some index set) induces a real *entropy function* $h : 2^{\mathcal{N}} \mapsto \mathbb{R}$ such that for any $\alpha \subseteq \mathcal{N}$,

$$h(\alpha) = H(Y_i, i \in \alpha)$$

is the joint Shannon entropy[2] of $(Y_i : i \in \alpha)$, which according to our notational conventions we will also write $H(Y_\alpha)$.

Let

$$\mathcal{H}[\mathcal{N}] \triangleq \mathbb{R}^{2^{|\mathcal{N}|}}$$

[2]We define $h(\alpha) = 0$ whenever α is an empty set.

be the $2^{|\mathcal{N}|}$-dimensional Euclidean space whose coordinates are indexed by subsets of $\mathcal{N}$. Thus, any element $g \in \mathcal{H}[\mathcal{N}]$ has coordinates $(g(\alpha), \alpha \subseteq \mathcal{N})$. Elements of $\mathcal{H}[\mathcal{N}]$ are called *rank functions*[3]. Clearly, entropy functions are rank functions.

Definition 8 (Entropic Functions): A rank function $h \in \mathcal{H}[\mathcal{N}]$ is

- *Entropic* if h is the entropy function of a set of $|\mathcal{N}|$ random variables. The set of entropic functions is denoted $\Gamma^*(\mathcal{N}) \subset \mathcal{H}[\mathcal{N}]$ [10]. When the index set $\mathcal{N}$ for the set of random variables is understood, we simply denote the set of entropic functions as Γ^*.
- *Weakly entropic* if there exists $c > 0$ such that $c \cdot h$ is entropic.
- *Almost entropic* if there exists a sequence of weakly entropic functions h^i such that

$$\lim_{i \to \infty} h^i = h.$$

  The set of almost entropic functions is $\bar{\Gamma}^*$.

For any rank function $g \in \mathcal{H}[\mathcal{N}]$, define the notations

$$g(\alpha \mid \beta) \triangleq g(\alpha \cup \beta) - g(\beta), \tag{13}$$

$$g(\alpha \wedge \beta) \triangleq g(\alpha) + g(\beta) - g(\alpha \cup \beta). \tag{14}$$

If g is in fact an entropy function induced by random variables $\{Y_i, i \in \mathcal{N}\}$, then $g(\alpha \mid \beta)$ is the usual conditional entropy $H(Y_\alpha \mid Y_\beta)$ and $g(\alpha \wedge \beta)$ is the usual mutual information $I(Y_\alpha; Y_\beta)$. We avoid the standard notation $I(\cdot;\cdot)$ since it hides the underlying entropy function, which will be critical in most of what we do.

The set Γ^* plays an important role in information theory. Characterisation of this set amounts to characterising every possible information inequality. Thus Γ^* essentially fixes the "laws" of information theory. However it turns out that Γ^* has a very complex structure and an explicit characterisation is still missing [11]. It has been proved that the closure $\bar{\Gamma}^*$ is a closed convex cone [10] and hence is more analytically manageable than Γ^*. For many applications, it is in fact sufficient to consider $\bar{\Gamma}^*$. However, it was proved in [12] that when $|\mathcal{N}| \geq 3$,

$$\Gamma^* \neq \bar{\Gamma}^*.$$

B. Existing Results

For a given network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$, let $\Gamma^*(\mathsf{P})$ and $\mathcal{H}[\mathsf{P}]$ respectively denote $\Gamma^*(\mathcal{S} \cup \mathcal{E})$ and $\mathcal{H}[\mathcal{S} \cup \mathcal{E}]$. Define the coordinate projection

$$\text{proj}_{\mathsf{P}} : \mathcal{H}[\mathsf{P}] \mapsto \chi(\mathsf{P})$$

such that for any $h \in \mathcal{H}[\mathsf{P}]$,

$$\text{proj}_{\mathsf{P}}[h](s) = h(s), \ \forall s \in \mathcal{S} \tag{15}$$

$$\text{proj}_{\mathsf{P}}[h](e) = h(e), \ \forall e \in \mathcal{E} \tag{16}$$

[3]This terminology comes from matroid theory and does not imply that such functions must be defined via ranks of linear operators – although such functions are rank functions by this definition.

Similarly, for any subset $\mathcal{R} \subseteq \mathcal{H}[\mathsf{P}]$,

$$\text{proj}_{\mathsf{P}}[\mathcal{R}] \triangleq \{\text{proj}_{\mathsf{P}}[h] : h \in \mathcal{R}\}. \tag{17}$$

Again, if the underlying network coding problem P is understood implicitly, we will simply use the notations $\text{proj}[h]$ and $\text{proj}[\mathcal{R}]$.

Consider any network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$. Define the following subsets of $\mathcal{H}[\mathsf{P}] \triangleq \mathcal{H}[\mathcal{S} \cup \mathcal{E}]$:

$$\mathcal{C}_{\mathsf{I}}(\mathsf{P}) \triangleq \left\{ h \in \mathcal{H}[\mathsf{P}] : h(\mathcal{S}) = \sum_{s \in \mathcal{S}} h(s) \right\}, \tag{18}$$

$$\mathcal{C}_{\mathsf{T}}(\mathsf{P}) \triangleq \left\{ h \in \mathcal{H}[\mathsf{P}] : h\left(s \mid \mathsf{in}(e)\right) = 0, \forall e \in \mathcal{E} \right\}, \tag{19}$$

$$\mathcal{C}_{\mathsf{D}}(\mathsf{P}) \triangleq \left\{ \begin{array}{l} h \in \mathcal{H}[\mathsf{P}] : h\left(s \mid \mathsf{in}(u)\right) = 0, \\ \hfill \forall s \in \mathcal{S}, u \in D(s) \end{array} \right\}. \tag{20}$$

The above subsets will be denoted by $\mathcal{C}_{\mathsf{I}}, \mathcal{C}_{\mathsf{T}}$ and $\mathcal{C}_{\mathsf{D}}$ respectively if the network coding problem P is understood implicitly. Consider a network code $\{Y_i, i \in \mathcal{S} \cup \mathcal{E}\}$ with induced entropy function $h \in \mathcal{H}[\mathcal{S} \cup \mathcal{E}]$. By Lemma 2.1 we see that $h \in \mathcal{C}_{\mathsf{I}}$ since the sources are mutually independent, and $h \in \mathcal{C}_{\mathsf{T}}$ due to deterministic transmission through the network. If the network code is zero-error, $h \in \mathcal{C}_{\mathsf{D}}$ follows from the decodability constraint (10).

The set of ϵ-achievable rate-capacity tuples can be characterised exactly as follows [1].

Theorem 2.1 (Yan, Yeung and Zhang – ϵ-achievable Region [1]): For a given network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$, a rate-capacity tuple $(\lambda, \omega) \in \chi(\mathsf{P})$ is ϵ-achievable if and only if

$$(\lambda, \omega) \in \mathsf{CL}(\text{proj}_{\mathsf{P}}\left[\overline{\text{con}}(\Gamma^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}}) \cap \mathcal{C}_{\mathsf{D}}\right]). \tag{21}$$

Inner and outer bounds for the 0-achievable region were also investigated in [13] using a similar framework as in [1], [4]. However, [13] allowed the use of variable length coding, where the amount of data traffic on a particular link is measured as the average number of transmitted bits.

In contrast, this paper studies the *worst case* scenario where the amount of traffic is measured by the maximum number of bits transmitted on a link (hence, it is sufficient to consider fixed-length codes). It is worth pointing out that when decoding error is not allowed, there is a significant difference between using the average or the maximum number of transmitted bits. For example, consider a source X compressed/encoded by an optimal uniquely-decodable code. The average length of resulting codeword is roughly equal to $H(X)$. However, for all uniquely-decodable codes, the maximum length of the encoded codeword must be at least $\log |\mathsf{SP}(X)|$, which can be much greater than $H(X)$ if X is heavily biased.

The following outer bound follows directly from Theorem 2.1, but was proved earlier in [2]

Corollary 2.1 (Yeung – Outer Bound [2]): If a rate-capacity tuple (λ, ω) is ϵ-achievable, then

$$(\lambda, \omega) \in \mathsf{CL}(\text{proj}_{\mathsf{P}}[\bar{\Gamma}^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}}]). \tag{22}$$

This bound (22) is not necessarily tight, since Γ^* is not closed and convex in general. Therefore,

$$\overline{\text{con}}(\Gamma^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}})$$

theoretically may be a proper subset of

$$\overline{\mathrm{con}}(\bar{\Gamma}^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}}) \overset{(a)}{=} \bar{\Gamma}^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}}$$

where (a) follows from that $\bar{\Gamma}^*$ is a closed and convex cone.

It is clear that if $(\lambda, \omega) \in \chi(\mathsf{P})$ is 0-achievable, then it is also ϵ-achievable and hence must satisfy the outer bound (22) in Corollary 2.1. In fact, it can be seen directly that (22) must be an outer bound for the set of 0-achievable rate-capacity tuples. Suppose $(\lambda, \omega) \in \chi(\mathsf{P})$ is 0-achievable. Then there exists a sequence of network codes $\{Y_f^n, f \in \mathcal{S} \cup \mathcal{E}\}$ with induced entropy functions h^n and positive constants c_n such that for all $e \in \mathcal{E}$ and $s \in \mathcal{S}$

$$\lim_{n\to\infty} c_n H(Y_e^n) \leq \lim_{n\to\infty} c_n \log |\mathsf{SP}(Y_e^n)| \leq \omega(e),$$
$$\lim_{n\to\infty} c_n H(Y_s^n) = \lim_{n\to\infty} c_n \log |\mathsf{SP}(Y_s^n)| \geq \lambda(s),$$

and that $h^n \in \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}}$. Consequently,

$$(\lambda, \omega) \in \mathsf{CL}(\mathsf{proj}_{\mathsf{P}}[\Gamma^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}}]).$$

The proof that (22) is an outer bound for ϵ-achievable tuples is similar. However, as vanishing error is allowed, Fano's inequality is invoked to ensure $\lim_{n\to\infty} c_n h^n \in \mathcal{C}_{\mathsf{D}}$.

In the next section, we deliver our first main result, namely that (22) is tight when the sources are colocated.

III. Tightness of Yeung's Outer Bound

The analytical challenges in characterising Γ^* (let alone its intersection with $\mathcal{C}_{\mathsf{I}}$ and $\mathcal{C}_{\mathsf{T}}$) may render Theorem 2.1 unattractive as a characterisation of the network coding capacity region. In this section we show that the more manageable bound of Corollary 2.1, which involves the closure of Γ^* is in fact tight when the sources are colocated, a notion that we make precise below in Definition 9. Our proof will use quasi-uniform random variables, discussed in III-A, which are a valuable tool in proving zero-error results. The proof of the main result, Theorem 3.1 is given in III-B.

Definition 9 (Colocated sources): Consider a network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$. Its sources are called *colocated* if

$$O(s) = O(s'), \quad \forall s, s' \in \mathcal{S}.$$

In other words, if a node has an access to any source s, it also has access to all the other sources.

Theorem 3.1 (Colocated sources): Consider a network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$ with colocated sources according to Definition 9. Then

1) A rate-capacity tuple (λ, ω) is 0-achievable if and only if it is ϵ-achievable.
2) The outer bound in Corollary 2.1 is tight.

We fail to prove the tightness of the outer bound in Corollary 2.1 when sources are not colocated. However, we will give evidence in III-C to support our conjecture that the outer bound should be tight in general.

A. Tools: Quasi-Uniform Random Variables

Before we prove Theorem 3.1 in Section III-B, we introduce key tools and intermediate results. In particular, the proof relies on the concept of quasi-uniform random variables, which are crucial for proving zero-error results.

Definition 10 (Quasi-Uniform Random Variables [11]): A set of random variables $\{X_i, i \in \mathcal{N}\}$ is called *quasi-uniform* if for any subset $\alpha \subseteq \mathcal{N}$, the random variable $X_\alpha \triangleq (X_i, i \in \alpha)$ is uniformly distributed over its support, or equivalently,

$$H(X_\alpha) = \log |\mathsf{SP}(X_\alpha)|.$$

Lemma 3.1: Suppose $\{A, B\}$ is quasi-uniform. Then one can construct a random variable W such that

$$H(W) = H(A \mid B),$$
$$H(A \mid B, W) = 0.$$

Sketch of proof: As $\{A, B\}$ is quasi-uniform, it can be proved from Definition 10 that for any $b \in \mathsf{SP}(B)$,

$$\Pr(A = a \mid B = b) = \begin{cases} 2^{-H(A|B)} & \text{if } \Pr(A = a, B = b) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Assume without loss of generality that

$$\{q(1, b), \ldots, q(2^{H(A|B)}, b)\}$$

is the set of all elements in $\mathsf{SP}(A)$ such that

$$\Pr(A = a \mid B = b) > 0.$$

Let W be a random variable such that for any (a, b) in $\mathsf{SP}(A, B)$,

$$\Pr(W = w \mid A = a, B = b) = \begin{cases} 1 & \text{if } a = q(w, b) \\ 0 & \text{otherwise.} \end{cases}$$

The lemma can then be verified directly. ■

Definition 11 (Quasi-Uniform Rank Functions): A rank function $h \in \mathcal{H}[\mathcal{N}]$ is called

- *Quasi-uniform* if h is the entropy function of a set of $|\mathcal{N}|$ quasi-uniform random variables.
- *Weakly quasi-uniform* if there exists $c > 0$ such that $c \cdot h$ is quasi-uniform;
- *Almost quasi-uniform* if there exists a sequence of weakly quasi-uniform rank functions h^i such that

$$\lim_{i \to \infty} h^i = h.$$

Lemma 3.2: If $h_1, h_2 \in \mathcal{H}[\mathcal{N}]$ are quasi-uniform, then their sum, defined for all $\mathcal{A} \subseteq \mathcal{N}$ as $h_1(\mathcal{A}) + h_2(\mathcal{A})$, is also quasi-uniform.

Proof: Suppose $A_\mathcal{N}$ and $B_\mathcal{N}$ are two independent sets of quasi-uniform random variables whose entropy functions are h_1 and h_2 respectively. It is straightforward to construct a new set of variables $X_\mathcal{N}$ with entropy function $h_1 + h_2$, via

$$X_i = (A_i, B_i), \quad \forall i \in \mathcal{N}.$$

The lemma follows, since $X_{\mathcal{N}}$, and hence $h_1 + h_2$, is quasi-uniform. ■

For any weakly entropic function h, [14] explicitly constructed a sequence of weakly quasi-uniform functions with limit h. It can be verified directly that this sequence of weakly quasi-uniform functions satisfies the same functional dependency constraints as h. Hence, we have the following proposition.

Proposition 3.1: For any weakly entropic rank function h, there exists a sequence of quasi-uniform random variables $\{U_i^\ell, i \in \mathcal{N}\}$ and positive numbers c_ℓ such that

1) For any $\alpha \subseteq \mathcal{N}$,

$$\lim_{\ell \to \infty} c_\ell H\left(U_\alpha^\ell\right) = h\left(\alpha\right). \tag{23}$$

2) If $h\left(k \mid \alpha\right) = 0$, then

$$H\left(U_k^\ell | U_\alpha^\ell\right) = 0, \quad \text{for all } \ell. \tag{24}$$

In other words, h is the limit of a sequence of weakly quasi-uniform functions f^ℓ where

$$h\left(k \mid \alpha\right) = 0 \implies f^\ell(k \mid \alpha) = 0.$$

In fact, Proposition 3.1 remains valid even if h is almost entropic.

Proposition 3.2: For any almost entropic rank function $h \in \bar{\Gamma}^*(\mathcal{N})$, there exists a sequence of quasi-uniform random variables $\{U_i^\ell, i \in \mathcal{N}\}$ and positive numbers c_ℓ such that (23) and (24) hold.

Proof: By [14], there exists a sequence of quasi-uniform random variables $\{U_i^\ell, i \in \mathcal{N}\}$ and positive numbers c_ℓ such for all $\alpha \subseteq \mathcal{N}$,

$$\lim_{\ell \to \infty} c_\ell H\left(U_\alpha^\ell\right) = h\left(\alpha\right). \tag{25}$$

The challenge however is that (24) may not hold if h is not weakly entropic (we only know that h is the limit of a sequence of weakly entropic functions). In the following, we will show how to modify the $\{U_i^\ell, i \in \mathcal{N}\}$ such that (24) indeed holds. First, notice that $\{U_i^\ell, i \in \mathcal{N}\}$ is quasi-uniform. Hence, for any $k \in \mathcal{N}$ and $\alpha \subseteq \mathcal{N}$, $\{U_k^\ell, U_\alpha^\ell\}$ is quasi-uniform. By Lemma 3.1, one can construct a random variable $W_{k,\alpha}^\ell$ such that

$$H\left(W_{k,\alpha}^\ell\right) = H(U_k^\ell \mid U_\alpha^\ell), \tag{26}$$

$$H\left(U_k^\ell \mid U_\alpha^\ell, W_{k,\alpha}^\ell\right) = 0. \tag{27}$$

Let

$$W_\Delta^\ell \triangleq \{W_{k,\alpha}^\ell, (k, \alpha) \in \Delta\}$$

where

$$\Delta \triangleq \{(k, \alpha) : h(k \mid \alpha) = 0, k \in \mathcal{N}, \alpha \subseteq \mathcal{N}\}$$

Then

$$
\begin{aligned}
0 &\leq \lim_{\ell\to\infty} c_\ell H\left(W_\Delta^\ell\right) \\
&\leq \lim_{\ell\to\infty} c_\ell \sum_{(k,\alpha)\in\Delta} H\left(W_{k,\alpha}^\ell\right) \\
&= \sum_{(k,\alpha)\in\Delta} \lim_{\ell\to\infty} c_\ell H(U_k^\ell \mid U_\alpha^\ell) \\
&= \sum_{(k,\alpha)\in\Delta} \lim_{\ell\to\infty} c_\ell (H(U_k^\ell, U_\alpha^\ell) - H(U_\alpha^\ell)) \\
&\overset{(a)}{=} \sum_{(k,\alpha)\in\Delta} \left(h\left(k,\alpha\right) - h\left(\alpha\right)\right) \\
&= 0
\end{aligned}
$$

where (a) follows from (25). Consequently,

$$\lim_{\ell\to\infty} c_\ell H\left(W_\Delta^\ell\right) = 0. \tag{28}$$

We now construct our new set of random variables, $\{V_i^\ell, i\in\mathcal{N}\}$ by defining

$$V_i^\ell \triangleq (U_i^\ell, W_\Delta^\ell), \quad \forall i\in\mathcal{N}.$$

It is obvious that $H(V_k^\ell \mid V_\alpha^\ell) = 0$ for all $(k,\alpha)\in\Delta$. Let f^ℓ be the entropy function of $V_\mathcal{N}^\ell$. Then by (25), for any $\beta\subseteq\mathcal{N}$,

$$
\begin{aligned}
h\left(\beta\right) &= \lim_{\ell\to\infty} c_\ell H\left(U_\beta^\ell\right) && (29) \\
&\leq \lim_{\ell\to\infty} c_\ell H\left(U_\beta^\ell, W_\Delta^\ell\right) && (30) \\
&\leq \lim_{\ell\to\infty} c_\ell H\left(U_\beta^\ell\right) + \lim_{\ell\to\infty} c_\ell H\left(W_\Delta^\ell\right) && (31) \\
&\overset{(b)}{=} \lim_{\ell\to\infty} c_\ell H\left(U_\beta^\ell\right) && (32) \\
&= h\left(\beta\right). && (33)
\end{aligned}
$$

where (b) is by (28). Consequently,

$$\lim_{\ell\to\infty} c_\ell f^\ell(\beta) = \lim_{\ell\to\infty} c_\ell H\left(U_\beta^\ell, W_\Delta^\ell\right) = h(\beta),\ \forall\beta\subseteq\mathcal{N}.$$

Since f^ℓ is weakly entropic and $f^\ell(k \mid \alpha) = 0$ for all $(k,\alpha)\in\Delta$, we can once again use Proposition 3.1 to construct a sequence of weakly quasi-uniform functions g^j such that $\lim_{j\to\infty} g^j = h$ and $g^j(k \mid \alpha) = 0$ for all $(k,\alpha)\in\Delta$. ■

B. Proof for Theorem 3.1

The first claim of Theorem 3.1 is that the ϵ-achievable and 0-achievable regions are equivalent when sources are colocated. For any network coding problem $\mathsf{P} = (\mathsf{G},\mathsf{M})$, it is clear that if $(\lambda,\omega)\in\chi(\mathsf{P})$ is 0-achievable, then it is

also ϵ-achievable and hence must satisfy the outer bound (22) in Corollary 2.1. Thus, to prove Theorem 3.1, it suffices to show that for colocated sources, the rate-capacity tuple $\mathrm{proj}_{\mathsf{P}}[h]$ is 0-achievable for all $h \in \bar{\Gamma}^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}}$.

Our proof technique is similar to that used in [1]. However, instead of constructing network codes from strongly typical sequences, we use quasi-uniform random variables. Codes constructed from typical sequences admit a small (but vanishing) error. However, as we shall see, codes constructed from quasi-uniform random variables can be carefully designed to ensure zero decoding error probability.

Consider a network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$ where all sources are colocated. Suppose

$$h \in \bar{\Gamma}^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{T}}.$$

Since h is almost entropic, Proposition 3.2 implies the existence of a sequence of quasi-uniform random variables

$$\{U_f^n, f \in \mathcal{S} \cup \mathcal{E}\} \tag{34}$$

and positive numbers c_n such that

$$\lim_{n\to\infty} c_n H\left(U_\alpha^n\right) = h\left(\alpha\right), \quad \forall \alpha \subseteq \mathcal{S} \cup \mathcal{E} \tag{35}$$

$$H\left(U_e^n \mid U_{\mathsf{in}(e)}^n\right) \overset{(a)}{=} 0, \quad \forall e \in \mathcal{E} \tag{36}$$

$$H\left(U_s^n \mid U_{\mathsf{in}(u)}^n\right) \overset{(b)}{=} 0, \quad \forall s \in \mathcal{S}, u \in D(s) \tag{37}$$

where (a) is due to $h \in \mathcal{C}_{\mathsf{T}}$ and (b) is due to $h \in \mathcal{C}_{\mathsf{D}}$. Furthermore, by Lemma 3.2 the sum of two quasi-uniform rank functions is quasi-uniform. Hence, we can assume without loss of generality that

$$\lim_{n\to\infty} c_n = 0 \tag{38}$$

and $H(U_1^n)$ grows unbounded. This assumption (38) will be used in the latter part when we construct a zero-error network code.

In the following, for each n, we will construct a zero-error network code $\{Y_f^n, f \in \mathcal{S} \cup \mathcal{E}\}$ from each set of quasi-uniform random variables $\{U_f^n, f \in \mathcal{S} \cup \mathcal{E}\}$ such that

$$\lim_{n\to\infty} c_n H\left(Y_s^n\right) = h(s) \tag{39}$$

$$\lim_{n\to\infty} c_n H\left(Y_e^n\right) \leq h(e) \tag{40}$$

and consequently, $\mathrm{proj}_{\mathsf{P}}[h]$ is 0-achievable and the outer bound is tight.

Code construction: For simplicity of notation, we will drop the superscript n in (34) and directly denote the set of quasi-uniform random variables by

$$\{U_f, f \in \mathcal{S} \cup \mathcal{E}\}.$$

Suppose first that the $U_s, s \in \mathcal{S}$ are mutually independent. The U_s are quasi-uniform and hence uniformly distributed over their support. Thus (36) holds and Lemma 2.1 implies that $\{U_f, f \in \mathcal{S} \cup \mathcal{E}\}$ in fact defines a network code. Furthermore, by (37), the decoding error probability is zero, and Theorem 3.1 is proved for this special case of independent sources.

Unfortunately, $\{U_s, s \in \mathcal{S}\}$ need not be mutually independent in general. To address this problem, we will modify these variables to satisfy the independency constraint. This is when we require all sources to be colocated.

Since the network G is acyclic, repeated application of (36) can be used to prove that

$$H\left(U_f \mid U_{\mathcal{S}}\right) = 0,\ \forall f \in \mathcal{S} \cup \mathcal{E}.$$

Hence, for any $e \in \mathcal{E}$, there exists functions G_e such that

$$U_e = G_e(U_{\mathcal{S}}).$$

Similarly, for any $e \in \mathcal{E}$, $s \in \mathcal{S}$ and $v \in D(s)$, there exists functions g_e and $g_{s,v}$ such that for all $u_{\mathcal{S}} \in \mathsf{SP}(U_{\mathcal{S}})$,

$$G_e(u_{\mathcal{S}}) \overset{(a)}{=} g_e(G_f(u_{\mathcal{S}}), f \in \mathsf{in}(e)), \quad \forall e \in \mathcal{E} \tag{41}$$

$$U_s \overset{(b)}{=} g_{s,v}(G_f(u_{\mathcal{S}}), f \in \mathsf{in}(v)),$$

$$\forall s \in \mathcal{S},\ v \in D(s). \tag{42}$$

where (a) follows from (36) and (b) from (37).

Definition 12 (Partition): Let $\mathcal{A} = \{1, 2, \ldots, |\mathcal{A}|\}$ be an index set. A *partition* of a set $\mathcal{X}$ into $|\mathcal{A}|$ partitions is a mapping

$$\Xi : \mathcal{A} \mapsto 2^{\mathcal{X}}$$

where $\Xi(i) \subseteq \mathcal{X}$ is the set of elements in partition $i \in \mathcal{A}$. If the $\Xi(i)$ are disjoint then the partition Ξ is called *disjoint*.

Definition 13 (Regular Partition Set): Let $U_s, s \in \mathcal{S}$ be random variables with supports $\mathsf{SP}(U_s)$. For $s \in \mathcal{S}$, define the index sets $\mathcal{A}_s \triangleq \{1, \ldots, |\mathcal{A}_s|\}$, where $|\mathcal{A}_s| \leq |\mathsf{SP}(U_s)|$, and let

$$\Xi_s : \mathcal{A}_s \mapsto 2^{\mathsf{SP}(U_s)}$$

be a disjoint partition of $\mathsf{SP}(U_s)$ into $|\mathcal{A}_s|$ subsets. We call the set of partitions $\{\Xi_s, s \in \mathcal{S}\}$ a *regular partition set* for $\{U_s, s \in \mathcal{S}\}$ if and only if for all $b_{\mathcal{S}} \in \mathcal{A}_{\mathcal{S}}$

$$\mathsf{SP}(U_{\mathcal{S}}) \cap \prod_{s \in \mathcal{S}} \Xi_s(b_s) \neq \emptyset. \tag{43}$$

Note that (43) is a non-trivial condition, since in general $\mathsf{SP}(U_{\mathcal{S}}) \neq \prod_s \mathsf{SP}(U_s)$ when the U_s are not necessarily independent.

We will now construct a zero-error network code $\{Y_f, f \in \mathcal{S} \cup \mathcal{E}\}$ from a regular partition set. Let $\{\Xi_s, s \in \mathcal{S}\}$ be a regular partition set according to Definition 13. By (43), for each $s \in \mathcal{S}$, there exists mappings

$$T_s : \prod_{i \in \mathcal{S}} \mathcal{A}_i \mapsto \mathsf{SP}(U_s), \tag{44}$$

such that for $s \in \mathcal{S}$ and $b_{\mathcal{S}} \in \prod_{i\in\mathcal{S}} \mathcal{A}_i$,

$$\big(T_1(b_{\mathcal{S}}), T_2(b_{\mathcal{S}}), \ldots, T_{|\mathcal{S}|}(b_{\mathcal{S}})\big) \in \mathsf{SP}(U_{\mathcal{S}}) \quad \text{and}$$
$$\big(T_1(b_{\mathcal{S}}), T_2(b_{\mathcal{S}}), \ldots, T_{|\mathcal{S}|}(b_{\mathcal{S}})\big) \in \Xi(b_1) \times \Xi(b_2) \times \cdots \times \Xi(b_{|\mathcal{S}|}).$$

We can write this more concisely as

$$(T_s(b_{\mathcal{S}}), s \in \mathcal{S}) \in \mathsf{SP}(U_{\mathcal{S}}) \cap \prod_{i\in\mathcal{S}} \Xi_i(b_i). \tag{45}$$

By (45),

$$T_s(b_{\mathcal{S}}) \in \Xi_s(b_s). \tag{46}$$

And as Ξ_s is a disjoint partition, b_s can be uniquely determined from $T_s(b_{\mathcal{S}})$.

Now let $\{Y_s, s \in \mathcal{S}\}$ be a set of mutually independent random variables such that for each $s \in \mathcal{S}$, Y_s is uniformly distributed over $\mathcal{A}_s$. Also, for each $s \in \mathcal{S}$, define auxiliary random variables Z_s such that

$$Z_s = T_s(Y_{\mathcal{S}}). \tag{47}$$

By (45) and (46), it is easy to see that

$$\mathsf{SP}(Z_{\mathcal{S}}) \subseteq \mathsf{SP}(U_{\mathcal{S}}), \tag{48}$$

$$H\left(Y_s \mid Z_s\right) = 0, \quad s \in \mathcal{S}. \tag{49}$$

Further define

$$Z_e \triangleq G_e(Z_{\mathcal{S}}), e \in \mathcal{E}. \tag{50}$$

It is now easy to see that[4]

$$\mathsf{SP}(Z_{\mathcal{S}\mathcal{E}}) \subseteq \mathsf{SP}(U_{\mathcal{S}\mathcal{E}}). \tag{51}$$

Following from (51), we have that

$$H\left(Z_k \mid Z_\alpha\right) = 0 \tag{52}$$

whenever $H\left(U_k \mid U_\alpha\right) = 0$ for some $k \in \mathcal{S} \cup \mathcal{E}$ and $\alpha \subseteq \mathcal{S} \cup \mathcal{E}$.

Let $Y_e = Z_e$ for all $e \in \mathcal{E}$. If $\mathsf{in}(e) \cap \mathcal{S} = \emptyset$, then

$$\begin{aligned} 0 &\stackrel{(i)}{=} H\left(U_e \mid U_{\mathsf{in}(e)}\right) \\ &\stackrel{(ii)}{=} H\left(Z_e \mid Z_{\mathsf{in}(e)}\right) \\ &\stackrel{(iii)}{=} H\left(Y_e \mid Y_{\mathsf{in}(e)}\right) \end{aligned}$$

[4]Recalling our convention to denote set union by juxtaposition.

where (i) follows from (36), (ii) from (52) and (iii) from the fact that $\mathsf{in}(e) \subseteq \mathcal{E}$. Now, suppose $\mathsf{in}(e) \cap \mathcal{S} \neq \emptyset$. As all sources are colocated in P, then $\mathsf{in}(e) \cap \mathcal{S} = \mathcal{S}$. In this case, application of (47) and (50) yields

$$0 \leq H(Y_e \mid Y_{\mathsf{in}}(e)) \leq H\left(Y_e \mid Y_{\mathcal{S}}\right) = H\left(Z_e \mid Y_{\mathcal{S}}\right) = 0.$$

Finally, for any $s \in \mathcal{S}$ and $u \in D(s)$,

$$\begin{aligned} 0 &= H\left(U_s \mid U_{\mathsf{in}(u)}\right) \\ &= H\left(Z_s \mid Z_{\mathsf{in}(u)}\right) \\ &= H\left(Z_s \mid Y_{\mathsf{in}(u)}\right) \\ &\overset{(a)}{=} H\left(Y_s \mid Y_{\mathsf{in}(u)}\right) \end{aligned}$$

where (a) follows from (49). Hence $\{Y_{\mathcal{S}\mathcal{E}}\}$ defines a zero-error network code for P. Furthermore, it is easy to see that

$$\log |\mathsf{SP}(Y_s)| = \log |\mathcal{A}_s| \tag{53}$$

$$\log |\mathsf{SP}(Y_e)| \leq \log |\mathsf{SP}(U_e)| = H\left(U_e\right). \tag{54}$$

Hence, (η, ζ) is 0-achievable where

$$\eta(s) = \log |\mathcal{A}_s|, \quad \forall s \in \mathcal{S}, \tag{55}$$

$$\zeta(e) = \log |U_e|, \quad \forall e \in \mathcal{E}. \tag{56}$$

The final ingredient of the proof is the following proposition which will be proved in Appendix A.

Proposition 3.3: Let $\{U_1, \ldots, U_S\}$ be a set of quasi-uniform random variables. If $H\left(U_1\right)$ is sufficiently large, then there exists at least one regular partition set $\{\Xi_s, s \in \mathcal{S}\}$ for $\{\mathsf{SP}(\mathcal{U}_s), s \in \mathcal{S}\}$ where

$$\mathcal{A}_s \triangleq \left\{1, \ldots, \frac{2^{H(U_s|U_1,\ldots,U_{s-1})}}{H\left(U_1, \ldots, U_s\right)^2}\right\}.$$

By Proposition 3.3 (and (38)), we can construct a sequence of zero-error network codes such that

$$\begin{aligned} &H\left(Y_s^n\right) = H\left(U_s^n \mid U_1^n, \ldots, U_{s-1}^n\right) - 2\log H\left(U_1^n, \ldots, U_s^n\right) \\ &H\left(Y_e^n\right) \leq H\left(U_e^n\right). \end{aligned}$$

Finally, from (35) and that $h \in \mathcal{C}_{\mathsf{I}}$, we can prove that

$$\begin{aligned} \lim_{n\to\infty} c_n H\left(Y_s^n\right) &= h(s), \\ \lim_{n\to\infty} c_n H\left(U_e^n\right) &\leq h(e) \end{aligned}$$

for all $s \in \mathcal{S}$ and $e \in \mathcal{E}$. Thus, $\mathsf{proj}_{\mathsf{P}}[h]$ is 0-achievable and Theorem 3.1 follows.

C. Generalisation – non-colocated sources

Assuming that all sources are colocated, we proved in the previous subsection that a rate-capacity tuple is ϵ-achievable if and only if it is indeed 0-achievable, and that the outer bound in Corollary 2.1 is indeed tight. We conjecture that the same outer bound remains tight even for sources are not colocated. In this subsection, we will give some arguments to justify our conjecture, which hinges on whether or not removing a zero-rate link can change the capacity of a certain modification of the original network.

Let $\mathsf{P} = (\mathsf{G}, \mathsf{M})$ where $\mathsf{G} = (\mathcal{V}, \mathcal{E})$ and $\mathsf{M} = (\mathcal{S}, O, D)$. We make no assumption that the sources are colocated. Now consider two variations on the network coding problem P.

Variation 1 – addition of a "super node": Let $\mathsf{P}^1 \triangleq (\mathsf{G}^\dagger, \mathsf{M}^1)$, where the underlying network $\mathsf{G}^\dagger = (\mathcal{V}^\dagger, \mathcal{E}^\dagger)$ is obtained from $\mathsf{G} = (\mathcal{V}, \mathcal{E})$ via inclusion of a "super node" v^* and links $f_s, s \in \mathcal{S}$,

$$\mathcal{V}^\dagger = \mathcal{V} \cup \{v^*\}, \tag{57}$$

$$\mathcal{E}^\dagger = \mathcal{E} \cup \{f_s, s \in \mathcal{S}\}, \tag{58}$$

such that $\mathsf{tail}(f_s) = v^*$ and $\mathsf{head}(f_s) = O(s)$. Here, the links f_s are just like an imaginary source edge.

The connection constraint M^1 is $(\mathcal{S}, O^1, D)$ where for all $s \in \mathcal{S}$,

$$O^1(s) = O(s) \cup \{v^*\}.$$

In other words, source s is available not only at source node $O(s)$ but also at the super node v^*.

Variation 2: Let $\mathsf{P}^2 \triangleq (\mathsf{G}^\dagger, \mathsf{M}^2)$, where the underlying network $\mathsf{G}^\dagger = (\mathcal{V}^\dagger, \mathcal{E}^\dagger)$ is the same as in P^1, but the connection constraint $\mathsf{M}^2 = (\mathcal{S}, O^2, D^2)$ is modified as follows:

$$O^2(s) = \{v^*\}, \quad \forall s \in \mathcal{S} \tag{59}$$

$$D^2(s) = D(s) \cup O(s). \tag{60}$$

Hence, all sources are available only at the super node v^*. In addition, source s is required to be reconstructed not only at the nodes in $D(s)$ (the sinks in the original multicast problem P) but also at the original source nodes $O(s)$ in P.

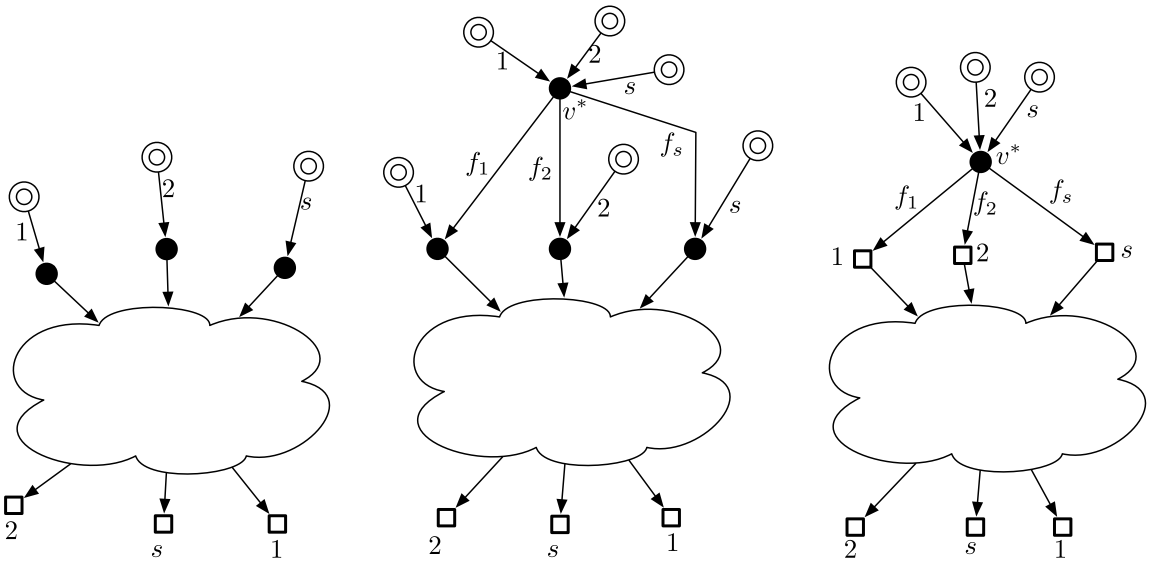
Figure 1 illustrates the differences between the original problem P and its two variations P^1 and P^2. In this figure, a sink node is denoted by an open square. Labels beside each sink node indicate the set of sources required for reconstruction. Sources are indicated by a double circle with an imaginary link (labeled with the source index) directed from it to the nodes where that source is available according to the connection constraint.

Given a rate-capacity tuple $(\lambda, \omega) \in \chi(\mathsf{P})$, let $T^1[\lambda, \omega]$ be a rate capacity tuple in $\chi(\mathsf{P}^1)$ (for the network coding problem P^1) such that

$$T^1[\lambda, \omega](s) = \lambda(s), \quad \forall s \in \mathcal{S}, \tag{61}$$

$$T^1[\lambda, \omega](e) = \omega(e), \quad \forall e \in \mathcal{E}, \tag{62}$$

$$T^1[\lambda, \omega](f_s) = 0, \quad \forall s \in \mathcal{S}. \tag{63}$$



(a) The original problem P. (b) The first variation P^1. (c) The second variation P^2.

Figure 1. Variations of a multicast problem P

In other words, the source rates and the hyperedge capacities remain the same for those elements existing in the original network, and the hyperedge capacities of the new links from the super-node v^* to each of the original source nodes are all zero.

Similarly, let $T^2[\lambda, \omega]$ be respectively the rate capacity tuple in $\chi(\mathsf{P}^2)$ (for the network coding problem P^2) such that

$$T^2[\lambda, \omega](s) = \lambda(s), \quad \forall s \in \mathcal{S}, \tag{64}$$

$$T^2[\lambda, \omega](e) = \omega(e), \quad \forall e \in \mathcal{E}, \tag{65}$$

$$T^2[\lambda, \omega](f_s) = \lambda(s), \quad \forall s \in \mathcal{S}. \tag{66}$$

Then, it is straightforward to prove the following:

1) $T^1[\lambda, \omega]$ is ϵ-achievable with respect to P^1 if and only if $T^2[\lambda, \omega]$ is ϵ-achievable with respect to P^2.
2) If (λ, ω) is ϵ-achievable with respect to P, then $T^1[\lambda, \omega]$ and $T^2[\lambda, \omega]$ are ϵ-achievable with respect to P^1 and P^2 respectively.

Conjecture 1: A rate-capacity tuple (λ, ω) is ϵ-achievable with respect to a network coding P if $T^1[\lambda, \omega]$ is ϵ-achievable with respect to the modified problem P^1.

The main difference between P and P^1 are the zero-capacity links f_s for $s \in \mathcal{S}$. At first sight it might be tempting to think that zero-capacity links cannot change the capacity region, and as a result that the conjecture is trivially true. However proving the conjecture is not straightforward. A zero-capacity link does not mean that absolutely nothing can be transmitted on the link. In fact according to the definitions, a finite amount of information (that

does not scale with n) could be transmitted along the link. Thus the links f_s can in fact be used (as long as their capacities vanish asymptotically) in any sequence of network codes achieving $T^1[\lambda, \omega]$. In fact there exists known examples where zero-capacity links can indeed modify the capacity of certain multi-terminal problems, in particular when there are correlated sources, or non-ergodic sources.

If Conjecture 1 does not hold, it is equivalent to saying that a link with "vanishing capacity" can indeed change the set of achievable tuples, even when all sources are independent and ergodic.

Theorem 3.2: Suppose Conjecture 1 holds. Then the outer bound in Corollary 2.1 is tight even when sources are not colocated.

Proof: Let $h \in \bar{\Gamma}^*(\mathsf{P})$. Define a rank function

$$g \in \mathcal{H}[\mathcal{S} \cup \mathcal{E}^\dagger]$$

such that for any $\beta \subseteq \mathcal{S} \cup \mathcal{E}^\dagger$

$$g(\beta) = h(\alpha_1 \cup \alpha_2) \tag{67}$$

where

$$\alpha_1 = \beta \setminus \{f_s, s \in \mathcal{S}\} \tag{68}$$

$$\alpha_2 = \{s \in \mathcal{S} : f_s \in \beta\}. \tag{69}$$

It is straightforward to prove that if

$$h \in \bar{\Gamma}^*(\mathsf{P}) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P}),$$

then

$$g \in \bar{\Gamma}^*(\mathsf{P}^2) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}^2) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}^2) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P}^2).$$

By Theorem 3.1, $\text{proj}_{\mathsf{P}^2}[g]$ is ϵ-achievable with respect to network coding problem P^2. As

$$T^2[\text{proj}_{\mathsf{P}}[h]] = \text{proj}_{\mathsf{P}^2}[g]$$

and is achievable with respect to P^2, $T^1[\text{proj}_{\mathsf{P}}[h]]$ is achievable with respect to P^1. By Conjecture 1, $\text{proj}_{\mathsf{P}}[h]$ is achievable with respect to P. Consequently, the outer bound in Corollary 2.1 is tight. ∎

IV. Linear Network Codes

In the previous section, the network codes were not subject to any constraints, other than those required by Definition 2 and that decoding error probabilities must be zero or vanishing, according to Definition 5 or Definition 6.

For the remainder of the paper, we will consider various subclasses of network codes which result from imposing different kinds of additional constraints.

To begin with, in this section we will study *linear network codes* [5], which have relatively low encoding and decoding complexities, making them more attractive for practical implementation.

Definition 14 (Linear network codes): Let

$$\{Y_f : f \ \in \mathcal{E} \cup \mathcal{S}\} \tag{70}$$

be a network code (according to Definition 2) for a problem P on a network $\mathsf{G} = (\mathcal{V}, \mathcal{E})$, with local encoding functions

$$\Phi \triangleq \{\phi_e : \ e \in \mathcal{E}\}.$$

The code is called *q-linear* (or simply linear) if it satisfies the following conditions:

1) For $s \in \mathcal{S}$, Y_s is a random row vector such that each of its entries is selected independently and uniformly over $GF(q)$.
2) All the local encoding functions are linear.

A network coding problem is said to be subject to a *q-linearity constraint* if only q-linear network codes are allowed.

Let the row vector Y_s have λ_s elements. Clearly, for a linear network code (70), one can construct a $\sum_{i \in \mathcal{S}} \lambda_i \times \lambda_s$ matrix G_s for any $s \in \mathcal{S}$ such that

$$Y_s = Y \times G_s. \tag{71}$$

where $Y \triangleq [Y_1 Y_2 \ldots Y_{|\mathcal{S}|}]$ is the length $\sum_{i \in \mathcal{S}} \lambda_i$ row vector obtained from the concatenation of the Y_i and $\times$ is the usual vector-matrix multiplication.

Similarly, as all local encoding functions are linear, for each $e \in \mathcal{E}$, there exists a $\sum_{i \in \mathcal{S}} \lambda_i \times \omega_e$ matrix G_e such that

$$Y_e = Y \times G_e. \tag{72}$$

Hence, the symbol transmitted on link Y_e is a length ω_e vector over $GF(q)$.

Following the nomenclature in [2], the matrices G_f, $f \in \mathcal{S} \cup \mathcal{E}$ will be called the *global encoding kernels*. They define the linear relation between Y_e (the message sent along edge e) and $\{Y_s, s \in \mathcal{S}\}$ (the symbols generated at the sources). It is easy to prove that

$$|\mathsf{SP}(Y_s)| = q^{\lambda_s} \tag{73}$$

$$|\mathsf{SP}(Y_e)| \leq q^{\omega_e}. \tag{74}$$

with equality holding in (74) when G_e has full column rank.

A sink node $u \in D(s)$ can uniquely decode a source Y_s if and only if it can solve for Y_s (but not necessarily all other Y_i, $i \in \mathcal{S} \setminus s$) from the following linear system with unknowns Y:

$$Y_e = Y \times G_e, \quad \forall e \in \mathsf{in}(u). \tag{75}$$

It is clear that if Y_s cannot be uniquely determined from (75), then there must be at least q solutions to (75) such that the values of Y_s in each solution are all different. We can easily show that with maximum likelihood decoding, the decoding error probability must be at least $1 - 1/q$. Consequently, for a network coding problem subject to

a q-linearity constraint, a rate-capacity tuple (λ, ω) is 0-achievable if and only if it is ϵ-achievable. Therefore, we will always assume that linear codes are zero-error codes.

As in Theorem 3.1 for general (possibly non-linear) network codes, a characterisation for the set of 0-achievable rate-capacity tuples subject to a q-linearity constraint can be obtained by using entropy functions. However as we shall see, the entropy functions for linear network codes will be constrained to be *representable* [15] in the sense of matroid theory.

Definition 15: A rank function $h \in \mathcal{H}[\mathcal{S} \cup \mathcal{E}]$ is called *q-representable* if there exists vector subspaces

$$\mathbb{U}_i, i \in \mathcal{S} \cup \mathcal{E}$$

over $GF(q)$ such that for all $\alpha \subseteq \mathcal{S} \cup \mathcal{E}$,

$$h(\alpha) = \dim \langle \mathbb{U}_i, i \in \alpha \rangle, \tag{76}$$

where $\dim \langle \mathbb{U}_i, i \in \alpha \rangle$ denotes the dimension of the smallest vector space containing all of the $\mathbb{U}_i, i \in \alpha$.

In an abstract sense, representable rank functions are similar to entropy functions, where the entropies of random variables are replaced by dimensions of vector spaces. It is well-known that representable functions are indeed entropy functions (and hence also polymatroidal) [10], [11]. Hence we will sometimes use the following conventions. For vector spaces $\mathbb{U}$ and $\mathbb{V}$, we will use

$$H(\mathbb{V}), \ H(\mathbb{U}, \mathbb{V}), \text{ and } H(\mathbb{U} \mid \mathbb{V})$$

to respectively denote

$$\dim \mathbb{V}, \ \dim \langle \mathbb{U}, \mathbb{V} \rangle, \text{ and } \dim \langle \mathbb{U}, \mathbb{V} \rangle - \dim \mathbb{U},$$

as if $\mathbb{U}$ and $\mathbb{V}$ were random variables.

Similar to Definition 11 we can define weakly- and almost-representable functions.

Definition 16 (Weakly/almost representable): A function h is called

- *weakly q-representable* if $c \cdot h$ is q-representable for some $c > 0$
- *almost q-representable* if it is the limit of a sequence of weakly q-representable functions.

We use $\Upsilon_q^*(\mathcal{S} \cup \mathcal{E})$ and $\bar{\Upsilon}_q^*(\mathcal{S} \cup \mathcal{E})$ to respectively denote the sets of q-representable and almost q-representable rank functions in $\mathcal{H}[\mathcal{S} \cup \mathcal{E}]$.

Theorem 4.1 (Achievability by linear codes): For any network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$ subject to a q-linearity constraint, a rate-capacity tuple (λ, ω) is 0-achievable if and only if

$$(\lambda, \omega) \in \mathsf{CL}\left(\mathrm{proj}_{\mathsf{P}}\left[\bar{\Upsilon}_q^*(\mathsf{P}) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P})\right]\right). \tag{77}$$

Theorem 4.1 (for linear network codes) is a counterpart to Theorem 3.1 (for general network codes codes). However, unlike in Theorem 3.1, Theorem 4.1 holds *even when the sources are not colocated.*

Before we proceed to prove Theorem 4.1 (which involves proving both an if-part and an only-if part), we will illustrate the main idea by proving the following special case of the if-part of Theorem 4.1.

Proposition 4.1: Consider a q-representable function $h \in \Upsilon_q^*(\mathsf{P})$ such that

$$h \in \mathcal{C}_{\mathsf{I}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P}) \tag{78}$$

and let $(\lambda, \omega) = \mathrm{proj}_{\mathsf{P}}[h]$. Then (λ, ω) is 0-achievable by q-linear network codes.

Proof: By Definition 15, there exists a collection of subspaces

$$\{\mathbb{V}_i, i \in \mathcal{S} \cup \mathcal{E}\}$$

over $GF(q)$ such that

$$h(\alpha) = \dim \langle \mathbb{V}_i, i \in \alpha \rangle.$$

As $h \in \mathcal{C}_{\mathsf{I}}(\mathsf{P})$,

$$h(\mathcal{S}) = \sum_{s \in \mathcal{S}} h(s).$$

Therefore,

$$\dim \langle \mathbb{V}_s, s \in \mathcal{S} \rangle = \sum_{s \in \mathcal{S}} \dim \mathbb{V}_s. \tag{79}$$

Similarly, as $h \in \mathcal{C}_{\mathsf{T}}(\mathsf{P}) \cap \bar{\Gamma}^*(\mathsf{P})$, we can prove that

$$h(e, \mathcal{S}) = h(\mathcal{S})$$

for all $e \in \mathcal{E}$. Consequently,

$$\mathbb{V}_e \subseteq \langle \mathbb{V}_s, s \in \mathcal{S} \rangle, \quad \forall e \in \mathcal{E}. \tag{80}$$

Let $k = \dim \langle \mathbb{V}_s, s \in \mathcal{S} \rangle$. By (79) and (80), we may assume without loss of generality that all elements in $\mathbb{V}_f$ are length k column vectors for $f \in \mathcal{S} \cup \mathcal{E}$. For each $s \in \mathcal{S}$, let M_s be a $k \times h(s)$ full column rank matrix such that the space spanned by its columns is equal to $\mathbb{V}_s$. Similarly, for each $e \in \mathcal{E}$, let M_e be a $k \times h(e)$ full column rank matrix such that the space spanned by its columns is equal to $\mathbb{V}_e$.

Let Z be a length k row vector such that each of its entries is independently and uniformly selected from $GF(q)$. Then Z, together with the matrices $\{M_f, f \in \mathcal{S} \cup \mathcal{E}\}$ induces a set of random variables

$$\{Y_f, f \in \mathcal{S} \cup \mathcal{E}\}$$

such that $Y_f \triangleq Z \times M_f$ for all $f \in \mathcal{S} \cup \mathcal{E}$. Setting $Y = [Y_s, s \in \mathcal{S}]$, and similarly $M = [M_s, s \in \mathcal{S}]$ we have

$$Y = Z \times M.$$

By (79), it is easy to see that M must be a $k \times k$ invertible matrix. Therefore

$$Z = Y \times M^{-1} \tag{81}$$

and consequently, $Y_f = Y \times G_f$ where $G_f = M^{-1} \times M_f$.

In the following, we will prove that $\{Y_f, f \in \mathcal{S} \cup \mathcal{E}\}$ is indeed a zero-error linear network code. To see this, first notice that Y_s is a random row vector of length $h(s)$ and all its entries are independently and uniformly distributed over $GF(q)$. Now, consider any $e \in \mathcal{E}$. As $h \in \mathcal{C}_{\mathsf{T}}$,

$$h(e, \mathsf{in}(e)) = h(\mathsf{in}(e)).$$

Therefore,

$$\dim \langle \mathbb{V}_e, \mathbb{V}_f, f \in \mathsf{in}(e) \rangle = \dim \langle \mathbb{V}_f, f \in \mathsf{in}(e) \rangle$$

or equivalently,

$$\mathbb{V}_e \subseteq \langle \mathbb{V}_f, f \in \mathsf{in}(e) \rangle.$$

Consequently, we can construct a matrix ψ_e such that

$$M_e = [M_f, f \in \mathsf{in}(e)] \times \psi_e$$

Therefore,

$$\begin{aligned} Y_e &= Z \times M_e \\ &= Z \times [M_f, f \in \mathsf{in}(e)] \times \psi_e \\ &= [Z \times M_f, f \in \mathsf{in}(e)] \times \psi_e \\ &= [Y_f, f \in \mathsf{in}(e)] \times \psi_e. \end{aligned} \tag{82}$$

The equation (82) clearly indicates that the local encoding functions are indeed linear.

Similarly, as $h \in \mathcal{C}_{\mathsf{D}}(\mathsf{P})$,

$$h(s, \mathsf{in}(u)) = h(\mathsf{in}(u))$$

for any $u \in D(s)$. We can once again construct a "decoding function" $\psi_{s,u}$ such that

$$Y_s = [Y_f, f \in \mathsf{in}(u)] \times \psi_{s,u}.$$

Hence, the receiver node u can uniquely decode Y_s from $\{Y_f, f \in \mathsf{in}(u)\}$ and the probability of decoding failure is zero. As $\{Y_f, f \in \mathcal{S} \cup \mathcal{E}\}$ is a zero-error linear network code, together with (73)-(74), (λ, ω) is 0-achievable by linear network codes. ■

Our next step is to prove that Proposition 4.1 holds, even when the function h in (76) is almost q-representable. Before we prove this extension, we will need a few basic results from linear algebra.

Lemma 4.1: Let $\mathbb{A}, \mathbb{B}$ be vector subspaces. Then there exists a subspace $\mathbb{C} \subseteq \mathbb{B}$ such that

$$\langle \mathbb{A}, \mathbb{C} \rangle = \langle \mathbb{A}, \mathbb{B} \rangle$$

$$\mathbb{A} \cap \mathbb{C} = \{\mathbf{0}\}.$$

Consequently, $H(\mathbb{C}) = H(\mathbb{B} \mid \mathbb{A}) = H(\mathbb{A}, \mathbb{B}) - H(\mathbb{A})$.

Using Lemma 4.1, for any subspace $\mathbb{A}$ of $\mathbb{B}$, one can easily construct a vector subspace $\mathbb{A}^*$ of $\mathbb{B}$ such that

$$\langle \mathbb{A}, \mathbb{A}^* \rangle = \mathbb{B}, \tag{83}$$

$$\mathbb{A} \cap \mathbb{A}^* = \{\mathbf{0}\}. \tag{84}$$

Any vector $u \in \mathbb{B}$ can be uniquely expressed as the sum of vectors $u_1 \in \mathbb{A}^*$ and $u_2 \in \mathbb{A}$. For notational simplicity, we use $T_{\mathbb{A}}(u)$ to denote u_1. Similarly,

$$T_{\mathbb{A}}(\mathbb{V}) \triangleq \{T_{\mathbb{A}}(u) : \ u \in \mathbb{V}\}.$$

Clearly, if $\mathbb{V}$ is a subspace, then so is $T_{\mathbb{A}}(\mathbb{V})$.

In general, $T_{\mathbb{A}}(u)$ depends on the specific choice of $\mathbb{A}^*$. However, all the results mentioned in this paper involving $T_{\mathbb{A}}$ remain valid for any legitimate choice of $\mathbb{A}^*$. The following lemma may be directly verified.

Lemma 4.2 (Properties): For any subspaces $\mathbb{A}, \mathbb{B}_1, \mathbb{B}_2$,

$$T_{\mathbb{A}}(\mathbb{B}_1) \cap \mathbb{A} = \{\mathbf{0}\} \tag{85}$$

$$\langle T_{\mathbb{A}}(\mathbb{B}_1), \mathbb{A} \rangle = \langle \mathbb{A}, \mathbb{B}_1 \rangle \tag{86}$$

$$H(T_{\mathbb{A}}(\mathbb{B}_1)) = H\left(\mathbb{B}_1 \mid \mathbb{A}\right) \tag{87}$$

$$\langle T_{\mathbb{A}}(\mathbb{B}_1), T_{\mathbb{A}}(\mathbb{B}_2) \rangle = T_{\mathbb{A}}(\langle \mathbb{B}_1, \mathbb{B}_2 \rangle) \tag{88}$$

Furthermore, if $\mathbb{B}_1 \subseteq \mathbb{B}_2$, then

$$T_{\mathbb{A}}(\mathbb{B}_1) \subseteq T_{\mathbb{A}}(\mathbb{B}_2).$$

Using $T_{\mathbb{A}}$, we can "transform" a set of subspaces

$$\{\mathbb{B}_1, \ldots, \mathbb{B}_n\}$$

into another set of subspaces

$$\{T_{\mathbb{A}}(\mathbb{B}_1), \ldots, T_{\mathbb{A}}(\mathbb{B}_n)\}$$

satisfying Lemmas 4.3 and 4.4 below, which are direct consequences of Lemma 4.2.

Lemma 4.3 (Conditioning):

$$H\left(T_{\mathbb{A}}(\mathbb{B}_i), i \in \alpha\right) = H\left(T_{\mathbb{A}}\langle \mathbb{B}_i, i \in \alpha \rangle\right) = H\left(\mathbb{B}_i, i \in \alpha \mid \mathbb{A}\right). \tag{89}$$

Lemma 4.4 (Preserving functional dependencies): If

$$H\left(\mathbb{B}_k \mid \mathbb{B}_i, i \in \alpha\right) = 0,$$

or equivalently, $\mathbb{B}_k \subseteq \langle \mathbb{B}_i, i \in \alpha \rangle$, then

$$H\left(T_{\mathbb{A}}(\mathbb{B}_k) \mid T_{\mathbb{A}}(\mathbb{B}_i), i \in \alpha\right) = 0.$$

Proposition 4.2: Consider any almost q-representable function $h \in \mathcal{H}[\mathcal{S} \cup \mathcal{E}]$. Let

$$\Delta \triangleq \{(k, \alpha) : k \in \mathcal{S} \cup \mathcal{E}, \alpha \subseteq \mathcal{S} \cup \mathcal{E} \text{ such that } h(k \mid \alpha) = 0\}.$$

Then there exists a sequence of weakly q-representable rank functions $h^\ell \in \mathcal{H}[\mathcal{S}\cup\mathcal{E}]$ such that

$$\lim_{\ell\to\infty} h^\ell = h \tag{90}$$

and

$$h^\ell(k \mid \alpha) = 0, \tag{91}$$

for all positive integers ℓ, and $(k,\alpha)\in\Delta$.

Proof: By definition, for any almost q-representable function h, there exists a sequence $\{\mathbb{U}_f^\ell, f \in \mathcal{S}\cup\mathcal{E}\}$, $\ell = 1,2,\dots$ of collections of subspaces over $GF(q)$ and $c_\ell > 0$ such that for any $\alpha \subseteq \mathcal{S}\cup\mathcal{E}$,

$$\lim_{\ell\to\infty} c_\ell H\left(\mathbb{U}_\alpha^\ell\right) = h\left(\alpha\right). \tag{92}$$

For every pair of $(k,\alpha)\in\Delta$, by using Lemma 4.1, we can construct a subspace $\mathbb{W}_{k,\alpha}^\ell$ such that

$$H\left(\mathbb{W}_{k,\alpha}^\ell\right) = H(\mathbb{U}_k^\ell,\mathbb{U}_\alpha^\ell) - H(\mathbb{U}_\alpha^\ell) \tag{93}$$

$$\mathbb{U}_k^\ell \subseteq \langle \mathbb{U}_\alpha^\ell, \mathbb{W}_{k,\alpha}^\ell\rangle. \tag{94}$$

Let $\mathbb{W}_\Delta^\ell = \langle \mathbb{W}_{k,\alpha}^\ell, (k,\alpha)\in\Delta\rangle$. By (92)-(93) and the fact that

$$h\left(k,\alpha\right) = h\left(\alpha\right), \quad \forall (k,\alpha)\in\Delta,$$

we have

$$\begin{aligned}
\lim_{\ell\to\infty} c_\ell H\left(\mathbb{W}_\Delta^\ell\right) &\le \lim_{\ell\to\infty} c_\ell \sum_{(k,\alpha)\in\Delta} H\left(\mathbb{W}_{k.\alpha}^\ell\right) \\
&= \sum_{(k,\alpha)\in\Delta} \lim_{\ell\to\infty} c_\ell H\left(\mathbb{W}_{k.\alpha}^\ell\right) \\
&= \sum_{(k,\alpha)\in\Delta} \left(h\left(k,\alpha\right) - h\left(\alpha\right)\right) \\
&= 0.
\end{aligned} \tag{95}$$

Define a new collection of subspaces

$$\mathbb{V}_f^\ell \triangleq \langle U_f^\ell, W_\Delta^\ell\rangle, \quad \forall f\in\mathcal{S}\cup\mathcal{E} \tag{96}$$

and let g^ℓ be the representable function induced by

$$\{\mathbb{V}_f^\ell, f\in\mathcal{S}\cup\mathcal{E}\}.$$

Obviously, for all $(k,\alpha)\in\Delta$, $\mathbb{V}_k^\ell$ is a subspace of $\mathbb{V}_\alpha^\ell$, or equivalently, $g^\ell(k,\alpha) = g^\ell(\alpha)$. By (95), using a similar argument as given in (29)-(33) in the proof of Proposition 3.2, we can also prove that

$$\lim_{\ell\to\infty} c_\ell g^\ell = h.$$

The proposition then follows by letting $h^\ell = c_\ell g^\ell$. ■

In the proof for Theorem 3.1, an extra step (via the introduction of a regular partition set) is taken to construct a network code from a set of quasi-uniform random variables. This extra step requires that all sources are colocated. As we shall see, when we construct linear codes from a set of subspaces, the colocated assumption is no longer needed.

Lemma 4.5: For any subspaces $\mathbb{A}, \mathbb{B}$,

$$H\left(\mathbb{A} \mid \mathbb{B}\right) = H\left(\mathbb{A} \mid \mathbb{A} \cap \mathbb{B}\right). \tag{97}$$

Proof: Direct verification. ■

Lemma 4.6: Let $\{\mathbb{V}_f, f \in \mathcal{S} \cup \mathcal{E}\}$ be a collection of subspaces. Then there exists a subspace $\mathbb{A}$ such that

$$H(T_{\mathbb{A}}(\mathbb{V}_s), s \in \mathcal{S}) = \sum_{s \in \mathcal{S}} H(T_{\mathbb{A}}(\mathbb{V}_s))$$

and

$$H\left(\mathbb{A}\right) = H\left(\mathbb{V}_{\mathcal{S}}\right) - \sum_{s \in \mathcal{S}} H\left(\mathbb{V}_s \mid \mathbb{V}_{\mathcal{S} \setminus s}\right).$$

Proof: Let

$$\mathbb{W}_s \triangleq \mathbb{V}_s \cap \langle \mathbb{V}_j, j \in \mathcal{S} \setminus s \rangle, \ \forall s \in \mathcal{S} \tag{98}$$

and $\mathbb{A} \triangleq \langle \mathbb{W}_s, s \in \mathcal{S} \rangle$. Then

$$\begin{aligned}
H\left(\mathbb{V}_{\mathcal{S}} \mid \mathbb{A}\right) &= H\left(\mathbb{V}_{\mathcal{S}} \mid \mathbb{W}_i, i \in \mathcal{S}\right) \\
&\geq \sum_{s \in \mathcal{S}} H\left(\mathbb{V}_s \mid \mathbb{W}_i, i \in \mathcal{S}, \mathbb{V}_k, k \neq s\right) \\
&\overset{(a)}{=} \sum_{s \in \mathcal{S}} H\left(\mathbb{V}_s \mid \mathbb{W}_s, \mathbb{V}_k, k \neq s\right) \\
&\overset{(b)}{=} \sum_{s \in \mathcal{S}} H\left(\mathbb{V}_s \mid \mathbb{V}_k, k \neq s\right) \\
&\overset{(c)}{=} \sum_{s \in \mathcal{S}} H\left(\mathbb{V}_s \mid \mathbb{W}_s\right) \\
&\geq \sum_{s \in \mathcal{S}} H\left(\mathbb{V}_s \mid \mathbb{W}_i, i \in \mathcal{S}\right) \\
&= \sum_{s \in \mathcal{S}} H\left(\mathbb{V}_s \mid \mathbb{A}\right) \\
&\geq H\left(\mathbb{V}_{\mathcal{S}} \mid \mathbb{A}\right)
\end{aligned}$$

where (a) follows from that $\mathbb{W}_k \subseteq \mathbb{V}_k$, (b) from that $\mathbb{W}_k \subseteq \langle \mathbb{V}_i, i \neq k \rangle$, and (c) from Lemma 4.5. As all the above inequalities are in fact equalities, we have

$$H\left(\mathbb{V}_{\mathcal{S}} \mid \mathbb{A}\right) = \sum_{s \in \mathcal{S}} H\left(\mathbb{V}_s \mid \mathbb{A}\right) = \sum_{s \in \mathcal{S}} H\left(\mathbb{V}_s \mid \mathbb{V}_k, k \neq s\right).$$

Finally, as $\mathbb{A} \subseteq \langle \mathbb{V}_s, s \in \mathcal{S} \rangle$,

$$H(\mathbb{A}) = H(\mathbb{V}_s, s \in \mathcal{S}) - H(\mathbb{V}_{\mathcal{S}} \mid \mathbb{A}) \tag{99}$$

$$= H(\mathbb{V}_s, s \in \mathcal{S}) - \sum_{s \in \mathcal{S}} H(\mathbb{V}_s \mid \mathbb{V}_k, k \neq s). \tag{100}$$

■

We now have all the elements required to prove Theorem 4.1.

Proof of Theorem 4.1 :

We begin with the if-part. Suppose

$$h \in \bar{\Upsilon}_q^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}}.$$

Using Proposition 4.2, we can construct a sequence of q-representable functions f^ℓ and $c_\ell > 0$ such that

$$\lim_{\ell \to \infty} c_\ell f^\ell = h$$

and each f^ℓ satisfies all of the same functional dependencies as h, i.e.

$$h(k \mid \alpha) = 0 \implies f^\ell(k \mid \alpha) = 0.$$

In particular,

$$f^\ell \in \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}}.$$

For each ℓ, by definition, there exists subspaces

$$\{\mathbb{U}_i^\ell, i \in \mathcal{S} \cup \mathcal{E}\}$$

over $GF(q)$ such that $f^\ell(\alpha) = H\left(\mathbb{U}_i^\ell, i \in \alpha\right)$.

Then by Lemma 4.6, there exists a subspace $\mathbb{A}^\ell$ such that

$$H(T_{\mathbb{A}^\ell}(\mathbb{U}_s^\ell), s \in \mathcal{S}) = \sum_{s \in \mathcal{S}} H(T_{\mathbb{A}^\ell}(\mathbb{U}_s^\ell)) \tag{101}$$

and

$$H\left(\mathbb{A}^\ell\right) = H\left(\mathbb{U}_s^\ell, s \in \mathcal{S}\right) - \sum_{s \in \mathcal{S}} H\left(\mathbb{U}_s^\ell \mid \mathbb{U}_{\mathcal{S} \setminus s}^\ell\right). \tag{102}$$

Let g^ℓ be the representable function induced by the subspaces

$$\{T_{\mathbb{A}^\ell}(\mathbb{U}_i^\ell), i \in \mathcal{S} \cup \mathcal{E}\}.$$

Then by (101), $g^\ell \in \mathcal{C}_{\mathsf{I}}$. As each $f^\ell \in \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}}$, by Lemma 4.4, $g^\ell \in \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}}$. By Proposition 4.1, $\mathrm{proj}_{\mathsf{P}}[g^\ell]$ is 0-achievable.

Due to (102),

$$\lim_{\ell \to \infty} c_\ell H\left(\mathbb{A}^\ell\right) = 0,$$

and hence,

$$\lim_{\ell \to \infty} c_\ell g^\ell = \lim_{\ell \to \infty} c_\ell f^\ell = h.$$

Thus, $\mathrm{proj}_{\mathsf{P}}[h]$ is also 0-achievable and the if-part of Theorem 4.1 is proved.

Now, we will prove the only-if part. Suppose $(\lambda, \omega) \in \mathsf{T}(\mathsf{P})$ is 0-achievable subject to a q-linearity constraint. Then there exists a sequence of zero-error linear network codes $\{Y_f^n, f \in \mathcal{S} \cup \mathcal{E}\}$ and positive constants c_n such that

$$\lim_{n\to\infty} c_n H(Y_e^n) \le \lim_{n\to\infty} c_n \log |\mathsf{SP}(Y_e^n)| \le \omega(e), \quad \forall e \in \mathcal{E},$$

$$\lim_{n\to\infty} c_n H(Y_s^n) = \lim_{n\to\infty} c_n \log |\mathsf{SP}(Y_s^n)| \ge \lambda(e), \quad \forall s \in \mathcal{S}.$$

Again, each set of random variables $\{Y_f^n, f \in \mathcal{S} \cup \mathcal{E}\}$ induces a q-representable function h^n such that $H\left(Y_f^n, f \in \alpha\right) = h^n(\alpha)$ for all $\alpha \subseteq \mathcal{S} \cup \mathcal{E}$. Since $\{Y_f^n, f \in \mathcal{S} \cup \mathcal{E}\}$ is a zero-error linear code, we have $h^n \in \bar{\Upsilon}_q^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}}$. Therefore,

$$(\lambda, \omega) \in \mathsf{CL}(\bar{\Upsilon}_q^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}})$$

and the theorem is proved. ■

V. Routing

Another class of network coding constraints that is of great practical importance is *routing*, which requires that network nodes perform only store-and-forward operations. We will consider two main cases. In Section V-A we consider networks where *all* nodes must perform routing. In Section V-B we consider heterogeneous networks consisting of both routing and network coding nodes.

A. Routing-only schemes

We first consider networks where the nodes are only able to perform routing. We will formalise what we mean by "routing" later, and proposed a generalisation. In such routing-based schemes, information is transmitted from the sources to the destinations via a collection of "routing subnetworks".

Definition 17 (Routing subnetworks): For any given network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$, a routing subnetwork is a subset $\mathcal{T}$ of $\mathcal{S} \cup \mathcal{E}$ such that

1) $|\mathcal{T} \cap \mathcal{S}| = 1$. Thus, $\mathcal{T}$ is associated with a source and we denote that unique source in $\mathcal{T} \cap \mathcal{S}$ by $\nu(\mathcal{T})$.
2) For any link $e \in \mathcal{T} \cap \mathcal{E}$, $\mathsf{in}(e) \cap \mathcal{T} \neq \emptyset$. In other words, either there exists another link $f \in \mathcal{T}$ such that

$$f \in \mathsf{in}(e),$$

or $\nu(\mathcal{T}) \in \mathsf{in}(e)$, i.e., the originating node of link e is a source node of $\nu(\mathcal{T})$. Hence, the subnetwork formed by the set of links in $\mathcal{T}$ is in fact "connected" and is "rooted" at $\nu(\mathcal{T})$.

A routing subnetwork is in fact a simple generalisation of the usual multicast trees used in networks with point-to-point links (i.e. the underlying network is a directed graph) for constructing a routing solution (where messages are being forwarded and relayed at intermediate nodes without coding). While it is sufficient to consider multicast trees in such networks, the concept of multicast trees does not extend naturally to wireless networks (where the

underlying network is a directed *hypergraph*). In particular, in our hypergraph model, links $\mathcal{E}$ are broadcast, i.e., the message sent over a link e can be received by more than one node. Therefore, it is not reasonable (and also not necessary) to insist that there is a unique path connecting a source to a sink. According to our definition of a multicast constraint, sources may also be available at more than one node. Therefore, the condition $s \in \mathsf{in}(e)$ means that there exists a node $u \in O(s)$ such that $u = \mathsf{tail}(e)$.

Definition 18 (Achievability): A rate-capacity tuple

$$(\lambda, \omega) \in \chi(\mathsf{P})$$

is called 0-achievable subject to a *routing constraint* if there exists a collection of routing subnetworks $\mathcal{T}_i$ and *subnetwork capacities* $c_i \geq 0$ such that

(R1) For any edge $e \in \mathcal{E}$,

$$\omega(e) \geq \sum_{i: e \in \mathcal{T}_i} c_i. \tag{103}$$

(R2) For any i and $u \in D(\nu(\mathcal{T}_i))$, there exists $e \in \mathcal{T}_i$ such that $u \in \mathsf{head}(e)$. In other words, the node u is on the routing subnetwork.

(R3) For any source $s \in \mathcal{S}$,

$$\lambda(s) = \sum_{i: \nu(\mathcal{T}_i) = s} c_i. \tag{104}$$

Clearly, these three conditions are not chosen arbitrarily but have a meaning in practice. Suppose (λ, ω) is 0-achievable subject to a routing constraint. This tuple corresponds to a zero-error routing solution defined as follows: Assume without loss of generality that $\lambda(s), \omega(e)$ and c_i are all positive integers. For each $s \in \mathcal{S}$, let the source message Y_s be a q-ary row vector of length $\lambda(s)$. For each i, one can use the routing subnetwork $\mathcal{T}_i$ to transmit c_i q-ary symbols of Y_s from the source nodes (which have access to the source Y_s) to all sink nodes $u \in D(s)$. By (R2), it is guaranteed that all sink nodes receive all $\lambda(s)$ q-ary symbols of Y_s and hence can decode Y_s. Furthermore, a link $e \in \mathcal{E}$ is used in the routing subnetwork $\mathcal{T}_i$ if $e \in \mathcal{T}_i$. Therefore,

$$\sum_{i: e \in \mathcal{T}_i} c_i$$

is the total number of q-ary symbols that have been transmitted on link e. Clearly, the rate-capacity tuple (λ, ω) is fit for this routing based solution.

In this routing solution, a source node does not perform any coding, except for partitioning a source message into several independent segments, and forwarding each segment via a routing subnetwork to the corresponding sink nodes. This corresponds to the usual concept of routing in networks consisting of point-to-point links. For successful decoding, a sink node must receive every segment of the source message from the required sources.

In the following, we consider a slight generalisation of the concept of routing, where source nodes can encode source messages into *correlated* segments (corresponding to intra-session coding). By doing so, we can weaken the conditions (R2) and (R3).

Definition 19 (Generalised routing constraint): A rate-capacity tuple (λ, ω) is called 0-achievable subject to a *generalised routing constraint* if there exists a collection of routing subnetworks $\mathcal{T}_i$ and *subnetwork capacities* $c_i \geq 0$ satisfying (R1) and the following condition:

(R2′) for any source $s \in \mathcal{S}$ and any sink node $u \in D(s)$,

$$\lambda(s) \leq \sum_{i:\mathsf{in}(u)\cap\mathcal{T}_i\neq\emptyset \text{ and } \nu(\mathcal{T}_i)=s} c_i. \tag{105}$$

Again, each 0-achievable tuple (λ, ω) subject to a generalised routing constraint is fit for a zero-error routing scheme as follows: Let Y_s be a q-ary row vector of length $\lambda(s)$. Instead of partitioning a source message Y_s into independent pieces, one can encode (e.g. using simple codes for erasure channels) Y_s into $\sum_{i:\nu(\mathcal{T}_i)=s} c_i$ q-ary symbols such that any $\lambda(s)$ encoded symbols can reconstruct Y_s with no error. These $\sum_{i:\nu(\mathcal{T}_i)=s} c_i$ symbols will be forwarded via the routing subnetworks $\mathcal{T}_i$ (where $\nu(\mathcal{T}_i) = s$) to sink nodes in $D(s)$. As before, all intermediate network nodes perform only store-and-forward operations. The condition (R2′) then guarantees that each sink node $u \in D(s)$ receives at least $\lambda(s)$ coded symbols of Y_s. Hence, the node can decode Y_s without error.

In the following, we will characterise the set of 0-achievable tuples subject to a (generalised) routing constraint using a similar framework as developed in Sections III and IV. Developing a characterisation within this same framework provides a convenient way to evaluate how a routing constraint may reduce the set of 0-achievable tuples. We should point out that we are not the first to characterise 0-achievable rate-capacity tuples subject to (generalised) routing constraints. In fact, if

$$|\mathsf{head}(e)| = 1, \quad \forall e \in \mathcal{E},$$

then the characterisation of 0-achievable rate-capacity tuples subject to (generalised) routing constraint can be obtained by solving variations of the fractional Steiner tree packing problem [16]. Our characterisation is however unified with the entropy function formulation used for network coding and highlights the differences (and similarities) between different characterisations with or without (generalised) routing constraints.

So far we have seen that entropy functions and representable entropy functions were the key ingredients in characterising the capacity regions for general network codes and for linear codes. For networks with routing constraints, we introduce *almost atomic functions* which in Theorem 5.1 below will provide the corresponding characterisation of the set of 0-achievable tuples.

Definition 20 (Atomic rank function): A rank function $h \in \mathcal{H}[\mathcal{S} \cup \mathcal{E}]$ is called *atomic* if there exists $\mathcal{T} \subseteq \mathcal{S} \cup \mathcal{E}$ such that

$$h(\beta) = \begin{cases} 1 & \text{if } \beta \cap \mathcal{T} \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \tag{106}$$

It is called *almost atomic* if it can be written

$$h = \sum_i c_i h^i$$

where for all i, $c_i \geq 0$ and h^i is atomic.

Let $\Gamma_{\mathsf{AA}}(\mathsf{P})$, or simply Γ_{AA}, be the set of all almost atomic rank functions in $\mathcal{H}[\mathcal{S} \cup \mathcal{E}]$. It can be easily proved that Γ_{AA} is a closed and convex cone contained in Γ^*. Thus, almost atomic rank functions are entropic.

Proposition 5.1: Let h be an atomic function such that there exists nonempty subset $\mathcal{T} \subseteq \mathcal{S} \cup \mathcal{E}$ and (106) holds. Then $\mathcal{T}$ is a routing subnetwork (for network coding problem P) if and only if

$$h \in \mathcal{C}_{\mathsf{T}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}).$$

Proof: We first prove the *only-if* part. Let $\mathcal{T}$ be a routing subnetwork. By definition, $|\mathcal{T} \cap \mathcal{S}| = 1$. It can be verified directly from definition that

$$h(\mathcal{S}) = \sum_{s \in \mathcal{S}} h(s).$$

Hence, $h \in \mathcal{C}_{\mathsf{I}}(\mathsf{P})$. It remains to prove that $h \in \mathcal{C}_{\mathsf{T}}(\mathsf{P})$.

For any $e \in \mathcal{E}$, if $e \notin \mathcal{T}$, then $h(\mathsf{in}(e), e) = h(\mathsf{in}(e))$ by (106). On the other hand, suppose $e \in \mathcal{T}$. Again as $\mathcal{T}$ is a routing subnetwork, $\mathsf{in}(e) \cap \mathcal{T}$ is nonempty. Thus,

$$h(\mathsf{in}(e), e) = 1 = h(\mathsf{in}(e)).$$

Hence, $h(\mathsf{in}(e), e) = h(\mathsf{in}(e))$ for all $e \in \mathcal{E}$. Consequently, $h \in \mathcal{C}_{\mathsf{T}}$. The only-if part follows.

Now, we will prove the *if*-part. Suppose $e \in \mathcal{T} \cap \mathcal{E}$. Then $h(e) = 1$. As $h \in \mathcal{C}_{\mathsf{T}}$,

$$h(e, \mathsf{in}(e)) = h(\mathsf{in}(e)) \tag{107}$$

and consequently $h(\mathsf{in}(e)) = 1$. By (106),

$$\mathcal{T} \cap \mathsf{in}(e) \neq \emptyset \tag{108}$$

and condition 2) of Definition 17 is satisfied.

As $h \in \mathcal{C}_{\mathsf{I}}$, it can verified directly that $|\mathcal{T} \cap \mathcal{S}| \leq 1$. Since the network G is acyclic and there are only finite number of links, there must exist at least one $s \in \mathcal{S}$ such that $s \in \mathcal{T}$. Consequently, $|\mathcal{T} \cap \mathcal{S}| = 1$. Hence condition 1) of Definition 17 is satisfied and the proposition follows. ■

Theorem 5.1 (Routing capacity): A rate-capacity tuple (λ, ω) is 0-achievable subject to a routing constraint if and only if

$$(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}_{\mathsf{P}}[\Gamma_{\mathsf{AA}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{I}}]).$$

Proof: We will first prove the *only-if* part. Suppose (λ, ω) is 0-achievable subject to a routing constraint. By Definition 18, there exists a collection of routing subnetworks $\mathcal{T}_i$ and nonnegative real numbers c_i such that conditions (R1) – (R3) hold.

By Proposition 5.1, each $\mathcal{T}_i$ is associated with an atomic rank function $h^i \in \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{I}} \cap \Gamma_{\mathsf{AA}}$ such that

$$h^i(\beta) = \begin{cases} 1 & \text{if } \beta \cap \mathcal{T}_i \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \tag{109}$$

For all sink nodes $u \in D(\nu(\mathcal{T}_i))$, (R2) implies that $h^i(\mathsf{in}(u)) = 1$. Hence, $1 = h^i(\mathsf{in}(u)) = h^i(\mathsf{in}(u), \nu(\mathcal{T}_i))$. On the other hand, if $s \neq \nu(\mathcal{T}_i)$, then $s \notin \mathcal{T}_i$ and $h^i(\mathsf{in}(u)) = h^i(\mathsf{in}(u), \nu(\mathcal{T}_i))$. Consequently, $h^i \in \mathcal{C}_{\mathsf{D}}$.

Let $h = \sum_i c_i h^i$. Since

$$h^i \in \Gamma_{\mathsf{AA}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{I}}$$

for all i, h is also in $\Gamma_{\mathsf{AA}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{I}}$. Finally, (R2) and (R3) imply that for any $s \in \mathcal{S}$

$$\lambda(s) = \sum_{i:\nu(\mathcal{T}_i)=s} c_i \tag{110}$$

$$= \sum_i c_i h^i(s) \tag{111}$$

$$= h(s). \tag{112}$$

Similarly, (R1) implies $\omega(e) \geq h(e)$ for all $e \in \mathcal{E}$. Thus, $(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}_{\mathsf{P}}[h])$ and the only-if part follows.

Now, we will prove the *if-part*. It is easy to prove that if (λ, ω) is 0-achievable subject to a routing constraint, then all tuples in $\mathsf{CL}(\lambda, \omega)$ are also 0-achievable. Therefore, it is sufficient to prove that $\mathrm{proj}_{\mathsf{P}}[h]$ is 0-achievable subject to a routing constraint for all

$$h \in \Gamma_{\mathsf{AA}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{I}}.$$

Since h is almost atomic, there exist atomic functions

$$h^i(\beta) = \begin{cases} 1 & \text{if } \beta \cap \mathcal{T}_i \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \tag{113}$$

such that $h = \sum_i c_i h^i$.

As each h^i is entropic (and hence polymatroidal) and c_i is nonnegative for all i,

$$\sum_i c_i h^i \in \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{I}}$$

implies that

$$h^i \in \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{I}}, \quad \forall i.$$

By Proposition 5.1, each $\mathcal{T}_i$ is in fact a routing subnetwork. Also, $h^i \in \mathcal{C}_{\mathsf{D}}$ implies that for any $u \in D(\nu(\mathcal{T}_i))$,

$$h(\mathsf{in}(u)) = h(\mathsf{in}(u), \nu(\mathcal{T}_i)) = 1.$$

This implies $\mathsf{in}(u) \cap \mathcal{T}_i \neq \emptyset$ and hence (R2) is satisfied.

For any $s \in \mathcal{S}$, and $u \in D(s)$,

$$h(s) = \sum_i c_i h^i(s) \tag{114}$$

$$\stackrel{(i)}{=} \sum_{i:\nu(\mathcal{T}_i)=s} c_i \tag{115}$$

where (i) follows from the fact that $h^i(s) = 0$ if $\nu(\mathcal{T}) \neq s$. Hence (R3) is satisfied. Condition (R1) can also be proved directly. The if-part is then proved. ■

Using a similar approach as in Theorem 5.1, we can also characterise the set of 0-achievable rate-capacity tuples subject to the generalised routing constraint.

Theorem 5.2 (Generalised routing capacity): Consider a network coding problem P. A rate-capacity tuple (λ, ω) is 0-achievable subject to the generalised routing constraint of Definition 19 if and only if

$$(\lambda, \omega) \in \mathsf{CL}(\text{proj}^*_{\mathsf{P}}[\Gamma_{\mathsf{AA}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{I}}]).$$

where

$$\text{proj}^*_{\mathsf{P}}[h](s) \triangleq \min_{u \in D(s)} h(s \wedge \mathsf{in}(u)) \tag{116}$$

$$\text{proj}^*_{\mathsf{P}}[h](e) \triangleq h(e). \tag{117}$$

Proof: Starting with the *only-if* part, suppose (λ, ω) is 0-achievable subject to the generalised routing constraint. By Definition 19, there exists a collection of routing subnetworks $\mathcal{T}_i$, and nonnegative constants c_i such that conditions (R1) and (R2′) hold. Each $\mathcal{T}_i$ is associated with a rank function $h^i \in \Gamma_{\mathsf{AA}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{I}}$ defined as in (109). Let

$$h = \sum_i c_i h^i.$$

Again, (R1) implies that

$$\omega(e) \geq h(e), \quad \forall e \in \mathcal{E}.$$

By (R2′), for any $s \in \mathcal{S}$ and $u \in D(s)$,

$$\begin{aligned} \lambda(s) &\leq \sum_{i : \mathsf{in}(u) \cap \mathcal{T}_i \neq \emptyset \text{ and } s \in \mathcal{T}_i} c_i & (118) \\ &\overset{(a)}{=} \sum_i c_i h^i(s \wedge \mathsf{in}(u)) & (119) \\ &= h(s \wedge \mathsf{in}(u)) & (120) \end{aligned}$$

where (a) follows from that

$$h^i(s \wedge \mathsf{in}(u)) = \begin{cases} 1 & \text{if } \mathsf{in}(u) \cap \mathcal{T}_i \neq \emptyset \text{ and } s \in \mathcal{T}_i \\ 0 & \text{otherwise.} \end{cases}$$

As (120) holds for all $u \in D(s)$, we have

$$\lambda(s) \leq \text{proj}^*_{\mathsf{P}}[h](s). \tag{121}$$

Thus, $(\lambda, \omega) \in \mathsf{CL}(\text{proj}^*_{\mathsf{P}}[h])$ and the only-if part follows.

Now, we will prove the *if*-part. Let $h \in \Gamma_{\mathsf{AA}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{I}}$ and $(\lambda, \omega) \in \mathsf{CL}(\text{proj}^*_{\mathsf{P}}[h])$. As before, we can construct a collection of functions h^i, routing subnetworks $\mathcal{T}_i$ and positive constants c_i such that $h = \sum_i c_i h^i$ and (113) holds. By definition,

$$\omega(e) \geq h(e), \; \forall e \in \mathcal{E}.$$

and for any $s \in \mathcal{S}$ and $u \in D(s)$,

$$\lambda(s) \leq \mathsf{proj}^*_{\mathsf{P}}[h](s) \tag{122}$$

$$\leq h(s \wedge \mathsf{in}(u)) \tag{123}$$

$$= \sum_{i:\mathsf{in}(u)\cap\mathcal{T}_i\neq\emptyset \text{ and } s\in\mathcal{T}_i} c_i. \tag{124}$$

Then both (R1) and (R2′) are satisfied and the result follows. ■

B. Heterogeneous networks: Partial routing constraints

In the previous section, we considered two varieties of routing schemes defined by routing subnetworks. In those schemes, each subnetwork is dedicated to sending a segment of data from a source to its respective sinks. Intermediate network nodes can only perform store-and-forward operations to forward the same data segment across a subnetwork. As only store-and-forward operations are performed, the computational requirements for intermediate nodes are relatively low. Despite this advantage, such routing-based schemes may suffer loss in throughput, as evidenced by the now famous example of the butterfly network [17]. In some cases, this loss can be significant.

In this section, we will consider more advanced schemes where some subsets of the intermediate nodes have sufficient computational resources to permit more sophisticated data processing in order to increase the throughput. Thus the network now consists of two types of nodes: routing nodes and network coding nodes. As demonstrated by the butterfly network example, there are known instances where the maximum possible throughput can in fact be achieved with only one network coding node, and all other nodes performing routing.

The aim of this section is to extend out methodology to such heterogeneous networks. As a first step, we need to clarify the concept of store-and-forward. Figure 2 is a subnetwork of G such that the node v is a "routing node" (where only store-and-forward operation is allowed). The node has two incoming links and one outgoing link. Suppose that v receives (b_0, b_1) from the incoming link e_1 and (b_2, b_3) from link e_2. A natural question is: If v can only perform store-and-forward operations, which types of outgoing message it can send?

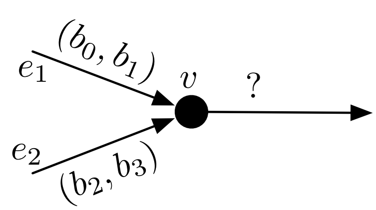


Figure 2. Partial routing contraint.

Naturally, we should allow the routing node v to send b_i for any $i = 0, 1, 2, 3$, but not $b_0 \oplus b_3$. The question however is if v can send $b_0 \oplus b_1$ which is a function of the incoming message from e_1?

In this paper, we will assume that v, as a routing node, is in fact permitted to perform intra-edge coding and send $b_0 \oplus b_1$. We do not allow inter-edge coding across different incoming links. Using this slightly generalised definition of routing, we can once again use the tools developed earlier to characterise 0-achievable/ϵ-achievable rate-capacity tuples for heterogeneous networks with "partial routing" constraints.

Definition 21 (Routing nodes): With respect to a network code $\{Y_f, f \in \mathcal{S} \cup \mathcal{E}\}$, an intermediate node v is said to be a *routing node* if for all outgoing links e of v (i.e., $\mathsf{tail}(e) = v$), there exist auxiliary random variables

$$\{Y_{f,e}, f \in \mathsf{in}(e)\}$$

such that

$$H\left(Y_e \mid Y_{f,e}, f \in \mathsf{in}(e)\right) = H\left(Y_{f,e}, f \in \mathsf{in}(e) \mid Y_e\right) = 0 \tag{125}$$

$$H\left(Y_{f,e} \mid Y_f\right) = 0,\ \forall f \in \mathsf{in}(e). \tag{126}$$

In other words, the outgoing message Y_e is formed by a set $\{Y_{f,e}, f \in \mathsf{in}(e)\}$ such that each element $Y_{f,e}$ is a function of the incoming message Y_f from the link f. *Routing links* are defined as outgoing links from a routing node.

Let $\varrho \subseteq \mathcal{E}$ be the set of all routing links. In other words, $e \in \varrho$ if and only if $\mathsf{tail}(e)$ is a routing node. We refer to ϱ as a *partial routing constraint*.

Definition 22 (Network code with partial routing constraints): A network code satisfying the partial routing constraint ϱ is a set of random variables

$$\{Y_i, i \in \mathcal{S} \cup \mathcal{E}\} \cup \{Y_{j,e}, e \in \varrho, j \in \mathsf{in}(e)\}.$$

such that $\{Y_i, i \in \mathcal{S} \cup \mathcal{E}\}$ is an ordinary network code according to Definition 2 and in addition, (125) and (126) hold for all $e \in \varrho$. We refer to such a code as a ϱ-network code

To go along with our definition of a network code with partial routing constraints, we need to update our definition of fitness.

Definition 23 (Fitness of a network code with partial routing): A rate-capacity tuple (λ, ω) is *fit* for a ϱ-network code $\{Y_i, i \in \mathcal{S} \cup \mathcal{E}\} \cup \{Y_{j,e}, e \in \varrho, j \in \mathsf{in}(e)\}$ if

$$\lambda(s) \le \log|\mathsf{SP}(Y_s)|, \forall s \in \mathcal{S} \tag{127}$$

$$\omega(e) \ge \sum_{f \in \mathsf{in}(e)} \log|\mathsf{SP}(Y_{f,e})|, \forall e \in \varrho \tag{128}$$

$$\omega(e) \ge \log|\mathsf{SP}(Y_e)|, \forall e \notin \varrho. \tag{129}$$

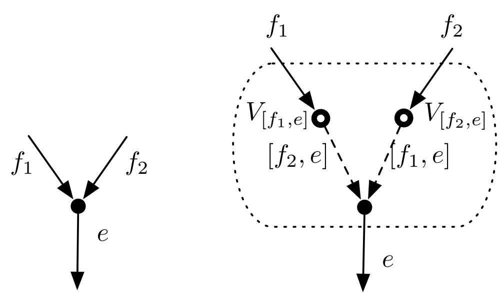
Note that we use (128) rather than (6) to highlight that the outgoing message Y_e will not be jointly compressed by the routing node. The set of 0-achievable and ϵ-achievable rate-capacity tuples subject to a partial routing constraint ϱ can be defined similar to Definitions 5 and 6.

Our approach for characterisation of the set of 0-achievable or ϵ-achievable rate-capacity tuples for ϱ-codes is to transform the problem with partial routing constraints into an equivalent unconstrained problem $(\mathsf{G}^\dagger, \mathsf{M})$.

Given a network coding problem (G,M) with partial routing constraint ϱ, define $\mathsf{G}^{\dagger} \triangleq (\mathcal{V}^{'}, \mathcal{E}^{'})$ as follows

1) Add new nodes:
$$\mathcal{V}^{'} = \mathcal{V} \cup \{V_{[j,e]}, e \in \varrho, j \in \mathsf{in}(e)\}.$$
2) Add new links:
$$\mathcal{E}^{'} = \mathcal{E} \cup \{[j,e], e \in \varrho, j \in \mathsf{in}(e)\}$$
such that
$$\mathsf{head}([j,e]) = \mathsf{tail}(e) \tag{130}$$
$$\mathsf{tail}([j,e]) = V_{[j,e]}. \tag{131}$$
3) Modifying existing link connections: For all $f \in \mathcal{E}$, the set $\mathsf{head}(f)$ is modified as
$$(\mathsf{head}(f) \setminus \{\mathsf{tail}(e) : e \in \varrho, f \in \mathsf{in}(e)\}) \cup \{V_{[f,e]} : e \in \varrho, f \in \mathsf{in}(e)\}.$$
In other words, if a link f was directed to a routing node $\mathsf{tail}(e)$ for some $e \in \varrho$, it will be redirected to the newly created node $V_{[f,e]}$.

Figure 3 is an example illustrating how to modify a network to remove the partial routing constraint. In this example, Figure 3(a) is one part of the network where $e \in \varrho$ is a routing link. Figure 3(b) shows how that part of the network is transformed. In the new network, we no longer impose any routing constraint.



(a) Original network. (b) Modified network.

Figure 3. Removing a routing contraint.

Theorem 5.3 (Network transformation): A rate-capacity tuple (λ,ω) is 0-achievable/ϵ-achievable with respect to a network coding problem (G,M) subject to a partial routing constraint ϱ if and only if there exists a 0-achievable/ϵ-achievable rate-capacity tuple $(\lambda^{'},\omega^{'})$ for the network coding problem $(\mathsf{G}^{\dagger},\mathsf{M})$ where
$$\lambda_s = \lambda_s^{'}, \quad \forall s \in \mathcal{S} \tag{132}$$
$$\omega_e = \omega_e^{'}, \quad \forall e \in \mathcal{E} \setminus \varrho \tag{133}$$
$$\omega_e = \sum_{f \in \mathsf{in}(e)} \omega^{'}_{[f,e]}, \quad \forall e \in \varrho. \tag{134}$$

Proof: By direct verification. ■

The construction of $\mathsf{G}^{\dagger}$ together with relationships between λ/ω and λ'/ω' remove the partial routing constraint by making the capacity of the links $e \in \varrho$ in $\mathsf{G}^{\dagger}$ sufficiently large such that network coding is never required at $\mathsf{head}(e)$, which has sufficient capacity to simply forward all of the incoming messages. Also note that the choice of $\omega'_{[f,e]}$ are free, apart from the constraints (134).

As a corollary of Theorem 5.3, all of the results obtained in the earlier sections can also be applied to network coding problems with partial routing constraints.

VI. Secure Network Codes

So far we have considered two classes of constraints on network codes. In Section IV we considered linear network codes, which may be attractive for practical implementation. In Section V we considered networks where some, or all of the nodes are constrained to perform only store-and-forward types of operations. Another important class of constraints to consider for network coding are motivated by security considerations. The objective is to determine the achievable network coding rates when we require secret transmission that is impervious to specified eavesdropping attacks.

Assume that there are $|\mathcal{R}|$ adversaries in the network. Adversary $r \in \mathcal{R}$ observes messages transmitted along links in the set $\mathcal{B}_r \subseteq \mathcal{E}$ and aims to reconstruct the set of sources indexed by $\mathcal{A}_r \subseteq \mathcal{S}$. We refer to $\mathsf{W} \triangleq \{(\mathcal{A}_r, \mathcal{B}_r), r \in \mathcal{R}\}$ as the *wiretapping pattern* of the network.

We will use the notation $\mathsf{P} = (\mathsf{G}, \mathsf{M}, \mathsf{W})$ to denote the network coding problem subject to a secrecy constraint, also referred to as a *secure network coding problem*. Here, G and M are as usual the network topology and the connection constraint. The secure communications objective is to transmit information over a network satisfying the multicast requirements while simultaneously ensuring that the eavesdroppers gain no information about their desired sources.

Before we characterise the set of 0-achievable/ϵ-achievable rate-capacity tuples for secure network coding, we need to point out a significant difference between ordinary and secure network codes. Without a secrecy constraint, the transmitted message on any network link e can be assumed without loss of generality to be a function of the source inputs and received messages available at the node $\mathsf{tail}(e)$. However, when secrecy constraints are enforced, it is usually necessary to encode messages *stochastically* to prevent an eavesdropper from learning any useful information about its desired sources.

Definition 24 (Stochastic network code): A *stochastic network code* is defined by a set of random variables

$$\{Y_f, f \in \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}\} \tag{135}$$

with entropy function h such that Y_s is uniformly distributed over its alphabet set for all $s \in \mathcal{S}$ and

$$h \in \mathcal{C}_{\mathsf{I}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P})$$

where

$$\mathcal{C}_{\mathsf{I}}(\mathsf{P}) \triangleq \begin{Bmatrix} h \in \mathcal{H}[\mathcal{S} \cup \mathcal{E} \cup \mathcal{V}] : \\ \quad h(\mathcal{S}, \mathcal{V}) = \sum_{s \in \mathcal{S}} h(s) + \sum_{u \in \mathcal{V}} h(u) \end{Bmatrix} \tag{136}$$

$$\mathcal{C}_{\mathsf{T}}(\mathsf{P}) \triangleq \begin{Bmatrix} h \in \mathcal{H}[\mathcal{S} \cup \mathcal{E} \cup \mathcal{V}] : \\ \quad h\left(s \mid \mathsf{in}(e), \mathsf{tail}(e)\right) = 0, \forall e \in \mathcal{E} \end{Bmatrix}. \tag{137}$$

In the definition, $\{Y_s, s \in \mathcal{S}\}$ and $\{Y_e, e \in \mathcal{E}\}$ are again the set of random sources (indexed by $s \in \mathcal{S}$) and the set of messages (transmitted on hyperedges $e \in \mathcal{E}$). The random variables $\{Y_u, u \in \mathcal{V}\}$ can be thought of as the *random keys* available at nodes $u \in \mathcal{V}$ for stochastic encoding. Specifically, each link $e \in \mathcal{E}$ is associated with a local encoding function such that

$$Y_e = \phi_e(Y_i, i \in \mathsf{in}(e), Y_{\mathsf{tail}(e)}). \tag{138}$$

Clearly, we have

$$H(Y_e \mid Y_i, i \in \mathsf{in}(e), Y_{\mathsf{tail}(e)})$$

and hence (137). Furthermore, we want to point out that the random keys $\{Y_u, u \in \mathcal{V}\}$ are *not* like the usual secret keys that are privately shared between nodes in Shannon-style secure communications. Instead, they are locally (and hence independently) generated at each node. In other words, there are *no* correlated or common keys shared privately between nodes in advance. Therefore, we will assume that $\{Y_f, f \in \mathcal{S} \cup \mathcal{V}\}$ are mutually independent, and require (136) to hold.

A. Weak Secrecy

Definition 25 (Weak secrecy): For a secure network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M}, \mathsf{W})$, a rate-capacity tuple (λ, ω) is called 0-achievable subject to a *weak secrecy* constraint if there exists a sequence of stochastic network codes

$$\Phi^n = \{Y_f^n : f \in \mathcal{E} \cup \mathcal{S} \cup \mathcal{V}\}$$

and positive normalising constants c_n such that

(S1) for all $e \in \mathcal{E}$ and $s \in \mathcal{S}$,

$$\lim_{n \to \infty} c_n \log |\mathsf{SP}(Y_e^n)| \leq \omega(e), \tag{139}$$

$$\lim_{n \to \infty} c_n \log |\mathsf{SP}(Y_s^n)| \geq \lambda(s). \tag{140}$$

(S2) for $s \in \mathcal{S}$ and $u \in D(s)$,

$$H(Y_s^n \mid Y_f^n, f \in \mathsf{in}(u)) = 0.$$

(S3) For all $r \in \mathcal{R}$,

$$\lim_{n \to \infty} c_n I(Y_{\mathcal{A}_r}^n; Y_{\mathcal{B}_r}^n) = 0. \tag{141}$$

Similarly, a rate capacity tuple (λ, ω) is called ϵ-achievable subject to a weak secrecy constraint if there exists a sequence of network codes

$$\Phi^n = \{Y_f^n : f \in \mathcal{E} \cup \mathcal{S} \cup \mathcal{V}\}$$

satisfying (S1) and (S3) and the following condition (S2$'$):

(S2$'$) for $s \in \mathcal{S}$ and $u \in D(s)$, there exists decoding functions $g_{s,u}^n$ such that

$$\lim_{n\to\infty} \Pr(Y_s^n \neq g_{s,u}^n(Y_f^n : f \in \mathsf{in}(u))) = 0.$$

The following theorem can be proved by using the same technique as in Corollary 2.1. For brevity, we state the theorem without proof.

Theorem 6.1 (Outer bound): Consider any secure network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M}, \mathsf{W})$ subject to a weak secrecy constraint. Let

$$\mathcal{C}_{\mathsf{D}}(\mathsf{P}) \triangleq \left\{ \begin{array}{r} h \in \mathcal{H}[\mathcal{S} \cup \mathcal{E} \cup \mathcal{V}] : h\left(s \mid \mathsf{in}(u)\right) = 0, \\ \forall s \in \mathcal{S}, u \in D(s) \end{array} \right\}, \tag{142}$$

$$\mathcal{C}_{\mathsf{S}}(\mathsf{P}) \triangleq \left\{ h \in \mathcal{H}[\mathcal{S} \cup \mathcal{E} \cup \mathcal{V}] : h\left(\mathcal{A}_r \wedge \mathcal{B}_r\right) = 0, \forall r \in \mathcal{R} \right\}. \tag{143}$$

If a rate-capacity tuple $(\lambda, \omega) \in \chi(\mathsf{P})$ is ϵ-achievable, then there exists

$$h \in \bar{\Gamma}^*(\mathcal{S} \cup \mathcal{E} \cup \mathcal{V}) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{S}}(\mathsf{P})$$

such that

$$\lambda(s) \leq h(s),\ \forall s \in \mathcal{S}, \tag{144}$$

$$\omega(e) \geq h(e),\ \forall e \in \mathcal{E}. \tag{145}$$

Or equivalently,

$$(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}_{\mathsf{P}}[\bar{\Gamma}^*(\mathcal{S} \cup \mathcal{E} \cup \mathcal{V}) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{S}}(\mathsf{P})]).$$

The condition (142) corresponds to the decoding constraint, requiring that any node $u \in D(s)$ can decode the source s with vanishingly small error. The condition (143) is the secrecy constraint, ensuring that an adversary can learn no information about the sources it is interested in.

Unlike in Theorem 3.1, we do not claim tightness of the outer bound even for colocated sources. This is because secure network nodes may locally generate random keys for the purpose of stochastic encoding. These keys, to a certain extent, behave like sources (with no corresponding sink nodes), and hence the colocated source condition fails, even if the actual sources are colocated.

B. Strong Secrecy

Weak secrecy requires that the amount of information leakage vanishes asymptotically after normalisation. In other words, the amount of information leakage is negligible (when compared with the size of the source messages). We can also consider a *strong secrecy constraint*, where we require the information leakage to be exactly zero.

Definition 26 (Strong secrecy): A rate-capacity tuple (λ, ω) is 0-achievable subject to a *strong secrecy* constraint if there exists a sequence of network codes

$$\Phi^n = \{Y_f^n : f \in \mathcal{E} \cup \mathcal{S} \cup \mathcal{V}\}$$

and positive normalising constants c_n satisfying (S1), (S2) and the following condition

(S3′) $I(Y_{\mathcal{A}_r}^n ; Y_{\mathcal{B}_r}^n) = 0$ for all n and $r \in \mathcal{R}$.

Similarly, it is ϵ-achievable subject to a strong secrecy constraint if the sequence of codes satisfies (S1), (S2′) and (S3′).

In general, it is very hard to characterise the set of achievable rate-capacity tuples subject to a strong secrecy constraint, even implicitly via entropy functions. However, under the additional constraint of linearity, the set of 0-achievable rate-capacity tuples can in fact be characterised implicitly via the use of representable functions.

Definition 27 (Strongly secure linear network codes): Let

$$\{Y_f : f \in \mathcal{E} \cup \mathcal{S} \cup \mathcal{V}\} \tag{146}$$

be a stochastic network code (according to Definition 24) for a secure network coding problem P on a network $\mathsf{G} = (\mathcal{V}, \mathcal{E})$. The code is called *q-linear* (or simply linear) if it satisfies the following conditions:

1) For $f \in \mathcal{S} \cup \mathcal{V}$, Y_f is a random row vector such that each of its entries is selected independently and uniformly over $GF(q)$.
2) For any $e \in \mathcal{E}$, there exists a linear function ϕ_e such that

$$Y_e = \phi_e(Y_{\mathsf{in}(e)}, Y_{\mathsf{tail}(e)}).$$

A network coding problem is said to be subject to a *q-linearity constraint* if only q-linear network codes are allowed.

Theorem 6.2 (Strongly secure linear network codes): Consider a secure network coding problem P where $|O(s)| = 1$ for all $s \in \mathcal{S}$. A rate-capacity tuple (λ, ω) is 0-achievable subject to q-linearity and strong secrecy if and only if

$$(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}_{\mathsf{P}}[\bar{\Upsilon}_q^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{S}}]).$$

Proof: We first prove the *only-if* part. Suppose (λ, ω) is 0-achievable subject to linearity and strong secrecy constraints. By definition, there exists a sequence of linear codes

$$\{Y_f^n, f \in \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}\}$$

with entropy function h^n and normalising constants $c_n > 0$ such that (S1), (S2) and (S3′) hold. By (S2) and (S3′),

$$h^n \in \Upsilon_q^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{S}}.$$

And hence, $c_n h^n \in \bar{\Upsilon}_q^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{S}}$. Consequently,

$$(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}_{\mathsf{P}}[\bar{\Upsilon}_q^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{S}}]).$$

and the only-if part follows.

Now let

$$h \in \bar{\Upsilon}_q^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{S}}.$$

To prove the *if*-part, it suffices to prove that $\mathrm{proj}_{\mathsf{P}}[h]$ is 0-achievable. As in the proof of Theorem 4.1, there exists a sequence of q-representable functions

$$h^n \in \Upsilon_q^* \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{I}} \tag{147}$$

and positive scalars c_n such that

$$h = \lim_{i \to \infty} c_n h^n. \tag{148}$$

For each n, h^n induces a zero-error linear network code

$$\{Y_i^n, i \in \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}\} \tag{149}$$

such that

$$H\left(Y_s^n\right) = h^n(s),\ \forall s \in \mathcal{S} \tag{150}$$

$$H\left(Y_e^n\right) = h^n(e),\ \forall e \in \mathcal{E}. \tag{151}$$

However, the linear network code (149) need not be strongly secure (i.e., $h^n \in \mathcal{C}_{\mathsf{S}}$). In the following, we will create from h^n another representable function g^n such that

$$\lim_{n \to \infty} c_n g^n = \lim_{n \to \infty} c_n h^n = h \tag{152}$$

$$g^n \in \Upsilon_q^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{S}}. \tag{153}$$

Since h^n is q-representable, there exists subspaces

$$\{\mathbb{V}_i^n, i \in \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}\}$$

such that for all $\alpha \subseteq \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}$,

$$h^n(\alpha) = \dim \langle \mathbb{V}_j^n, j \in \alpha \rangle.$$

For each $r \in \mathcal{R}$ and $s \in \mathcal{A}_r$, we define

$$\mathbb{W}_{r,s}^n \triangleq \mathbb{V}_s^n \cap \langle \mathbb{V}_f^n, f \in \mathcal{B}_r \rangle. \tag{154}$$

Then by direct verification,

$$H\left(\mathbb{W}_{r,s}^n\right) = h^n(s \wedge \mathcal{B}_r)$$

and hence $\lim_{n \to \infty} c_n H\left(\mathbb{W}_{r,s}^n\right) = 0$.

Let

$$\mathbb{W}_s^n \triangleq \langle \mathbb{W}_{r,s}^n, r \in \mathcal{R} \text{ and } s \in \mathcal{A}_r \rangle.$$

By Lemma 4.1, for every $s \in \mathcal{S}$, there exists a subspace $\mathbb{U}_s^n$ of $\mathbb{V}_s^n$ such that

$$\dim \mathbb{V}_s^n = \dim \mathbb{U}_s^n + \dim \mathbb{W}_s^n, \tag{155}$$

$$\{\mathbf{0}\} = \mathbb{U}_s^n \cap \mathbb{W}_s^n. \tag{156}$$

For any $s \in \mathcal{S}$, let $O(s)$ be the unique source node where the s^{th} source is available. Let

$$\mathbb{U}_u^n = \langle \mathbb{V}_u^n, \mathbb{W}_s^n, O(s) = u \rangle, \quad \forall u \in \mathcal{V} \tag{157}$$

$$\mathbb{U}_e^n = \mathbb{V}_e^n, \quad \forall e \in \mathcal{E}. \tag{158}$$

On the other hand,

$$\mathbb{U}_s^n \cap \langle \mathbb{V}_f^n, f \in \mathcal{B}_r \rangle \subseteq \mathbb{V}_s^n \cap \langle \mathbb{V}_f^n, f \in \mathcal{B}_r \rangle \tag{159}$$

$$= \mathbb{W}_{r,s}^n. \tag{160}$$

As

$$\mathbb{U}_s^n \cap \mathbb{W}_{r,s}^n = \{\mathbf{0}\},$$

we have

$$\mathbb{U}_s^n \cap \langle \mathbb{V}_f^n, f \in \mathcal{B}_r \rangle = \{\mathbf{0}\}.$$

Since $h^n \in \mathcal{C}_{\mathsf{I}}$,

$$\dim \langle \mathbb{V}_s, s \in \mathcal{S} \rangle = \sum_{s \in \mathcal{S}} \dim \mathbb{V}_s. \tag{161}$$

As $\mathbb{U}_s \subseteq \mathbb{V}_s$ for $s \in \mathcal{S}$,

$$\dim \langle \mathbb{U}_s^n, s \in \mathcal{S} \rangle = \sum_{s \in \mathcal{S}} \dim \mathbb{U}_s^n. \tag{162}$$

Let g^n be the representable function induced by

$$\{\mathbb{U}_f^n, f \in \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}\}.$$

Then, it can be directly verified that

1) $g^n \in \Upsilon_q^* \cap \mathcal{C}_{\mathsf{I}} \cap \mathcal{C}_{\mathsf{T}} \cap \mathcal{C}_{\mathsf{D}} \cap \mathcal{C}_{\mathsf{S}}$ where g^n is the rank function induced by $\{Y_f^n, f \in \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}\}$, and
2) $\lim_{n \to \infty} c_n g^n = h$.

Consequently, $\text{proj}_{\mathsf{P}}[g^n]$, and also $\text{proj}_{\mathsf{P}}[c_n g^n]$ and $\text{proj}_{\mathsf{P}}[h]$, are 0-achievable subject to the two constraints. ■

C. Secret Sharing

In secret sharing [18], a secret is shared among a set of users $\mathcal{N}$ where each user holds a component of the secret. The main objective is to ensure that only specified legitimate subgroups of users (indexed by a subset $\mathcal{A}$ of $\mathcal{N}$) can successfully decode the secret. All other illegitimate subgroups of users should receive no information about the secret. The collection of all legitimate subsets Ω is called the *access structure* of the secret sharing problem.

We can reformulate a secret sharing problem as a secure network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M}, \mathsf{W})$. In this secure network coding problem, there is only one source (the secret) which is only available at the source node u^*. There are $|\mathcal{N}|$ intermediate nodes, each of which represents a user. The transmitted message an intermediate node (or a user) received from the source corresponds to the component of the secret that it holds. There are $|\Omega|$ sink nodes indexed by $\{v_\alpha, \alpha \in \Omega\}$. The sink node v_α is connected to nodes (or users) $i \in \alpha$ and aims to reconstruct the secret. We also assume that each $\beta \notin \Omega$, is associated with an eavesdropper who can wiretap the set of edges $\{e_i, i \in \beta\}$. The secrecy constraint implies that all illegitimate subgroups of users have no information about the secret.

Mathematically, the secure network coding problem is defined as follows.

1) $\mathsf{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{u^*\} \cup \mathcal{N} \cup \{v_\alpha, \alpha \in \Omega\}$ and $\mathcal{E} = \{e_i, f_i, i \in \mathcal{N}\}$;
2) for any $i \in \mathcal{N}$, $\mathsf{tail}(e_i) = u^*$, $\mathsf{head}(e_i) = \{i\}$, $\mathsf{tail}(f_i) = i$ and $\mathsf{head}(f_i) = \{v_\alpha : i \in \alpha\}$;
3) $\mathsf{M} = (\mathcal{S}, O, D)$ where (i) $\mathcal{S} = \{1\}$, (ii) $O(1) = \{u^*\}$ and (iii) $D(1) = \{v_\alpha, \alpha \in \Omega\}$;
4) $\mathsf{W} = \{(1, e_i, i \in \beta) : \beta \subseteq \mathcal{N} \text{ and } \beta \notin \Omega\}$.

By translating a secret sharing problem to a secure network coding problem, the results obtained in this paper can be applied to secret sharing.

VII. Challenges in characterising achievable tuples

Characterising the set of achievable rate-capacity tuples for a network coding problem is generally very hard. So far, there are only a limited number of scenarios where the sets of 0-achievable/ϵ-achievable rate-capacity tuples have been explicitly determined. One scenario is when there is only one source, $|\mathcal{S}| = 1$ and no partial routing constraint or secrecy constraints. In this case, the set of achievable rate-capacity tuples is explicitly characterised by the cut-set bound [19]. If a secrecy constraint is additionally imposed, the set of achievable tuples can still be determined if *(i)* all links have unit capacity and *(ii)* the eavesdropper is t-*uniform* in the sense that an eavesdropper can wiretap any t links in the network [20], [6], [21]. In both cases, linear codes are optimal.

It is natural to wonder whether there is any hope that wide classes of network coding problems could have simple, explicit characterisations. In the following two subsections, we will show that even in some very simple scenarios, finding the set of achievable rate-capacity tuples can be extremely hard. The first scenario will be an incremental multicast. The second scenario is a secure multicast.

A. Incremental Multicast

In a *incremental multicast* problem, sources are totally ordered such that a receiver who wants to reconstruct source s is also required to reconstruct all other sources i for $i < s$. Here, the symbol $<$ is defined with respect to the total ordering of the sources. Incremental multicast is common in multimedia transmission, where data such as video or audio may be encoded into multiple layers. A layer can only be used for reconstruction at a receiver if all its previous layers are also available. This leads directly to an incremental multicast problem.

We will construct the simplest case of a incremental multicast problem involving two layers, colocated at the same source node. Hence, there are two types of receivers: those which request source $1'$, and those which request both

sources ($1'$ and $2'$). Even for such a simple setup, we will show that determining the set of achievable rate-capacity tuples can be as hard as solving any network coding problem in general.

To prove our claim, we consider any ordinary network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$, where sources may or may not be colocated. We will transform P into a two-layer incremental multicast problem $\mathsf{P}^\dagger = (\mathsf{G}^\dagger, \mathsf{M}^\dagger)$ and prove in Theorem 7.1 that determining the set of achievable tuples in the incremental multicast problem $\mathsf{P}^\dagger$ is at least as hard as determining the outer bound of Corollary 2.1 for the problem P.

The network $\mathsf{G}^\dagger = (\mathcal{V}^\dagger, \mathcal{E}^\dagger)$ is obtained from its subgraph G by adding nodes and hyperedges

$$\mathcal{V}^\dagger \triangleq \mathcal{V} \cup \{\gamma_s, \tau_{s,u},\ s \in \mathcal{S}, u \in D(s)\} \cup \{\psi, \phi, \eta, \eta^*\}$$

$$\mathcal{E}^\dagger \triangleq \mathcal{E} \cup \{a_s, b_s, c_s, s \in \mathcal{S}\} \cup \{d_{s,u}, s \in \mathcal{S}, u \in D(s)\}$$

with connections

$$\mathsf{tail}(a_s) = \phi \tag{163}$$

$$\mathsf{head}(a_s) = O(s) \cup \{\gamma_s, \eta^*\}. \tag{164}$$

$$\mathsf{tail}(b_s) = \phi \tag{165}$$

$$\mathsf{head}(b_s) = \{\gamma_s, \eta, \psi\} \cup \bigcup_{j \neq s} \{\tau_{j,v}, v \in D(j)\} \tag{166}$$

$$\mathsf{tail}(c_s) = \gamma_s \tag{167}$$

$$\mathsf{head}(c_s) = \{\eta, \eta^*\} \cup \{\tau_{s,v}, v \in D(s)\} \tag{168}$$

$$\mathsf{tail}(d_{s,u}) = u \tag{169}$$

$$\mathsf{head}(d_{s,u}) = \tau_{s,u}, \tag{170}$$

for all $s \in \mathcal{S}, u \in D(s)$.

In addition to augmenting the network, we also need to define the connection requirement $\mathsf{M}^\dagger = (\mathcal{S}^\dagger, O^\dagger, D^\dagger)$. In our two-layer incremental multicast problem there are two sources indexed by

$$\mathcal{S}^\dagger \triangleq \{1', 2'\}.$$

All the sources are colocated at the node ϕ, i.e.,

$$O^\dagger(1^{'}) = O^\dagger(2^{'}) = \phi.$$

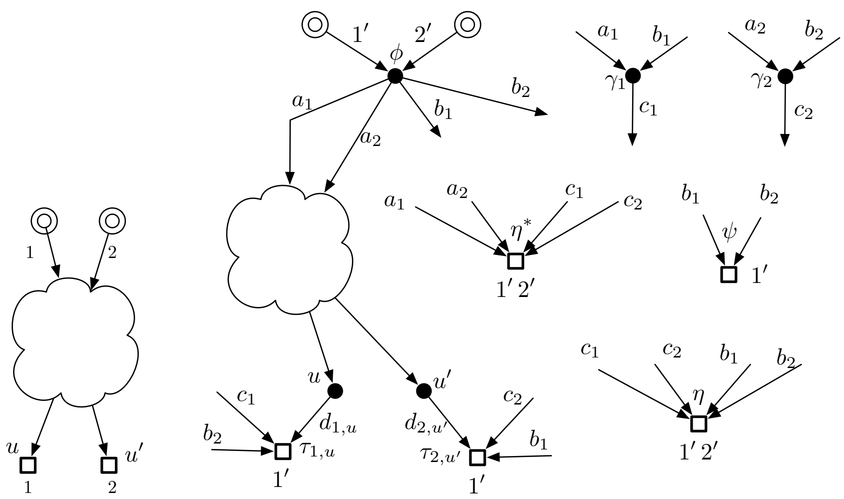
Finally, the destination location mapping $D^\dagger$ is defined as

$$D^\dagger(1^{'}) = \{\tau_{s,u}, s \in \mathcal{S}, u \in D(s)\} \cup \{\eta, \eta^*, \psi\},$$

$$D^\dagger(2^{'}) = \{\eta, \eta^*\}.$$

Figure 4 exemplifies how to convert an ordinary network coding problem P (which has two sources) into a two layers incremental multicast problem. Here, a source will be denoted by a double circle, and a sink by an open

square. The label beside a source or a sink denotes the index of the sources which are available or are required at the node. Note that in the figure, the sink nodes u and u' in the original problem P are no longer sink nodes in the incremental multicast problem $\mathsf{P}^\dagger$. Any rate-capacity tuple $(\lambda,\omega)\in\chi(\mathsf{P})$ for P induces another rate-capacity



(a) Network G (b) Network $\mathsf{G}^\dagger$

Figure 4. Transformation from G to $\mathsf{G}^\dagger$.

tuple

$$(\lambda^\dagger,\omega^\dagger)\triangleq T^\dagger(\lambda,\omega)$$

in $\chi(\mathsf{P}^\dagger)$ for $\mathsf{P}^\dagger$ such that

$$\omega^\dagger(e)=\omega(e), \tag{171}$$

$$\omega^\dagger(a_s)=\omega^\dagger(b_s)=\omega^\dagger(c_s)=\lambda(s), \tag{172}$$

$$\omega^\dagger(d_{s,u})=\lambda(s), \tag{173}$$

$$\lambda^\dagger(1^{'})=\lambda^\dagger(2^{'})=\sum_{s\in\mathcal{S}}\lambda(s). \tag{174}$$

where $e\in\mathcal{E}, s\in\mathcal{S}$ and $u\in D(s)$.

Theorem 7.1: Let (λ,ω) be a rate-capacity tuple in $\chi(\mathsf{P})$. Then the following two claims are valid.

1) If $T^\dagger(\lambda,\omega)\in\mathsf{CL}(\mathsf{proj}_{\mathsf{P}^\dagger}[\bar{\Gamma}^*(\mathsf{P}^\dagger)\cap\mathcal{C}_\mathsf{I}(\mathsf{P}^\dagger)\cap\mathcal{C}_\mathsf{T}(\mathsf{P}^\dagger)\cap\mathcal{C}_\mathsf{D}(\mathsf{P}^\dagger)])$, then

$$(\lambda,\omega)\in\mathsf{CL}(\mathsf{proj}_{\mathsf{P}}[\bar{\Gamma}^*(\mathsf{P})\cap\mathcal{C}_\mathsf{I}(\mathsf{P})\cap\mathcal{C}_\mathsf{T}(\mathsf{P})\cap\mathcal{C}_\mathsf{D}(\mathsf{P})]).$$

2) If a rate-capacity tuple (λ,ω) for P is 0-achievable, then $T^\dagger(\lambda,\omega)$ is 0-achievable with respect to $\mathsf{P}^\dagger$.

Proof: See Appendix B ■

Using a similar arguments as in Theorem 7.1, we can also prove the following theorem, whose proofs we omit for brevity.

Theorem 7.2: Let (λ, ω) be a rate-capacity tuple in $\chi(\mathsf{P})$. Then the following two claims are valid.

1) If $T^{\dagger}(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}_{\mathsf{P}^{\dagger}}[\bar{\Upsilon}_q^*(\mathsf{P}^{\dagger}) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}^{\dagger}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}^{\dagger}) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P}^{\dagger})])$, then
$$(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}_{\mathsf{P}}[\bar{\Upsilon}_q^*(\mathsf{P}) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P})]).$$
2) If (λ, ω) is 0-achievable with respect to P subject to the q-linearity constraint, then $T^{\dagger}(\lambda, \omega)$ is also 0-achievable with respect to $\mathsf{P}^{\dagger}$, subject to the q-linearity constraint.

Corollary 7.1 (Colocated sources): Suppose all the sources are colocated in P. Then

1) (λ, ω) is 0-achievable with respect to P if and only if $T^{\dagger}(\lambda, \omega)$ is also 0-achievable with respect to $\mathsf{P}^{\dagger}$.
2) (λ, ω) is 0-achievable with respect to P subject to the q-linearity constraint if and only if $T^{\dagger}(\lambda, \omega)$ is also 0-achievable with respect to $\mathsf{P}^{\dagger}$ subject to the same linearity constraint.

Proof: A direct consequence of Theorems 3.1, 4.1, 7.1 and 7.2. ■

In [4], a specific network coding problem P was proposed, such that all sources are colocated and that determining the set of ϵ-achievable rate-capacity tuple is at least as hard as determining the set of all information inequalities. Furthermore, it was also proved that linear codes are not optimal[5]. Therefore, by Corollary 7.1, we can directly prove the following proposition.

Proposition 7.1: There exists a two-layer incremental multicast network coding problem $\mathsf{P}^{\dagger}$ such that

1) Characterising the set of achievable rate-capacity tuples for a two-layer incremental network is in general no simpler than determining the set of all information inequalities.
2) Linear codes are not optimal.

B. Secure Multicast

In this subsection, we consider another scenario, very simple secure network coding problem with only one source. We will again show that solving the resulting secure network coding problem can be as hard as solving a general multi-source unconstrained network coding problem. Our approach is essentially the same as that used in the previous subsection for the incremental multicast. We will construct a simple single-source secure network coding problem $\mathsf{P}^{\ddagger} = (\mathsf{G}^{\ddagger}, \mathsf{M}^{\ddagger}, \mathsf{W})$ from an ordinary network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$. We will then show that solving the so-constructed secure network coding problem is as hard as solving the original network coding problem.

Construct the network $\mathsf{G}^{\ddagger}$ in $\mathsf{P}^{\ddagger}$ by adding nodes and hyperedges
$$\mathcal{V}^{\ddagger} \triangleq \mathcal{V} \cup \{\psi_s, \gamma_s, \theta_{s,u}, \tau_{s,u},\ s \in \mathcal{S}, u \in D(s)\} \cup \{\phi, \eta\}$$
$$\mathcal{E}^{\ddagger} \triangleq \mathcal{E} \cup \{a_s, b_s, c_s, e_s, s \in \mathcal{S}\} \cup \{d_{s,u}, w_{s,u}, s \in \mathcal{S}, u \in D(s)\}$$

[5] Linear codes are not optimal in the sense that there exists a 0-achievable rate-capacity tuple (λ, ω) for P which is not achievable when subject to the additional linearity constraint.

with link connections:

$$
\begin{aligned}
\mathsf{tail}(a_s) &= \phi \\
\mathsf{head}(a_s) &= O(s) \cup \{\eta\} \cup \{\gamma_s\} \\
\mathsf{tail}(b_s) &= \phi \\
\mathsf{head}(b_s) &= \{\gamma_s, \eta\} \cup \bigcup_{j \neq s} \{\theta_{j,i}, i \in D(j)\} \\
\mathsf{tail}(c_s) &= \gamma_s \\
\mathsf{head}(c_s) &= \{\psi_s\} \\
\mathsf{tail}(d_{s,u}) &= u \\
\mathsf{head}(d_{s,u}) &= \tau_{s,u} \\
\mathsf{tail}(e_s) &= \phi \\
\mathsf{head}(e_s) &= \{\psi_s\} \cup \{\tau_{s,i}, i \in D(s)\}.
\end{aligned}
$$

for all $s \in \mathcal{S}, u \in D(s)$.

The connection requirement $\mathsf{M}^\ddagger = (\mathcal{S}^\ddagger, O^\ddagger, D^\ddagger)$ is defined as follows:

$$
\begin{aligned}
O(1^{'}) &\triangleq \phi \\
D(1^{'}) &\triangleq \{\tau_{s,u}, s \in \mathcal{S}, u \in D(s)\} \cup \{\psi_s, s \in \mathcal{S}\} \cup \{\eta\} \\
\mathsf{W} &\triangleq \{(\mathcal{A}_1, \mathcal{B}_1), (\mathcal{A}_2, \mathcal{B}_2)\}
\end{aligned}
$$

where

$$
\begin{aligned}
\mathcal{A}_1 &= \mathcal{A}_2 = 1' \\
\mathcal{B}_1 &= \{a_s, s \in \mathcal{S}\} \\
\mathcal{B}_2 &= \{b_s, s \in \mathcal{S}\}.
\end{aligned}
$$

Figure 5 exemplifies how to convert an unconstrained network coding problem P into a single-source secure network coding problem.

As before, for any rate-capacity tuple $(\lambda, \omega) \in \chi(\mathsf{P})$, we define a tuple $(\lambda^\ddagger, \omega^\ddagger) \triangleq T^\ddagger(\lambda, \omega) \in \chi(\mathsf{P}^\ddagger)$ as follows:

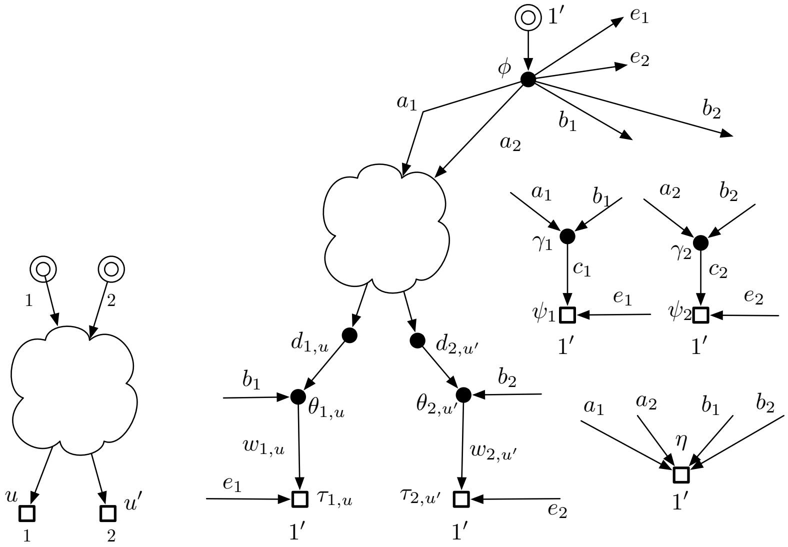
$$\omega^\ddagger(e) = \omega(e), \tag{175}$$

$$\omega^\ddagger(a_s) = \omega^\ddagger(b_s) = \omega^\ddagger(c_s) = \lambda(s), \tag{176}$$

$$\omega^\ddagger(d_{s,u}) = \lambda(s), \tag{177}$$

$$\omega^\ddagger(e_s) = \sum_{i \in \mathcal{S} \setminus \{s\}} \lambda(i) \tag{178}$$

$$\lambda^\ddagger(1^{'}) = \sum_{s \in \mathcal{S}} \lambda(s) \tag{179}$$



(a) Network G (b) Network $\mathsf{G}^\ddagger$

Figure 5. Transformation from G to $\mathsf{G}^\ddagger$.

for all $e \in \mathcal{E}, s \in \mathcal{S}$ and $u \in D(s)$.

Theorem 7.3: Let (λ, ω) be a rate-capacity tuple in $\chi(\mathsf{P})$. Then the following two claims are valid.

1) If $T^\ddagger(\lambda, \omega) \in \mathsf{CL}(\text{proj}(h^\ddagger))$ for some

$$h^\ddagger \in \bar{\Gamma}^*(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{I}(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{T}(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{D}(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{S}(\mathsf{P}^\ddagger).$$

then

$$(\lambda, \omega) \in \mathsf{CL}(\text{proj}(\bar{\Gamma}^*(\mathsf{P}) \cap \mathcal{C}_\mathsf{I}(\mathsf{P}) \cap \mathcal{C}_\mathsf{T}(\mathsf{P}) \cap \mathcal{C}_\mathsf{D}(\mathsf{P}))).$$

2) If a rate-capacity tuple (λ, ω) for P is 0-achievable, then $T^\ddagger(\lambda, \omega)$ is 0-achievable with respect to $\mathsf{P}^\ddagger$ subject to the strong secrecy constraint.

Proof: See Appendix C ■

The following theorem is the counterpart of Theorem 7.2. Again, its proof will be omitted.

Theorem 7.4: Let (λ, ω) be a rate-capacity tuple in $\chi(\mathsf{P})$. Then the following two claims are valid.

1) If $T^\ddagger(\lambda, \omega) \in \mathsf{CL}(\text{proj}_{\mathsf{P}^\ddagger}[\bar{\Gamma}^*(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{I}(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{T}(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{D}(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{S}(\mathsf{P}^\ddagger)])$, then

$$(\lambda, \omega) \in \mathsf{CL}(\text{proj}_\mathsf{P}[\bar{\Upsilon}_q^*(\mathsf{P}) \cap \mathcal{C}_\mathsf{I}(\mathsf{P}) \cap \mathcal{C}_\mathsf{T}(\mathsf{P}) \cap \mathcal{C}_\mathsf{D}(\mathsf{P})]).$$

2) If (λ, ω) is 0-achievable with respect to P subject to the q-linearity constraint, then $T^\ddagger(\lambda, \omega)$ is also 0-achievable with respect to $\mathsf{P}^\ddagger$, subject to the strong secrecy and q-linearity constraint.

Corollary 7.2 (Counterpart of Corollary 7.1): Suppose all the sources are colocated in P. Then

1) (λ, ω) is 0-achievable with respect to P if and only if $T^\ddagger(\lambda, \omega)$ is also 0-achievable with respect to $\mathsf{P}^\ddagger$ subject to the strong secrecy constraint.

2) (λ, ω) is 0-achievable with respect to P subject to the q-linearity constraint if and only if $T^{\ddagger}(\lambda, \omega)$ is also 0-achievable with respect to $\mathsf{P}^{\ddagger}$ subject to the strong secrecy and q-linearity constraint.

Proposition 7.2 (Counterpart of Proposition 7.1): There exists a single source secure multicast network coding problem $\mathsf{P}^{\ddagger}$ such that

1) Characterising the set of achievable rate-capacity tuples for $\mathsf{P}^{\ddagger}$ is in general no simpler than determining the set of all information inequalities.
2) Linear codes may not be optimal.

Remark: In [6], it was proved that in the single-source case, if all links have equal capacity and the eavesdroppers' capability is limited by the total number of links it can wiretap, then linear network codes are optimal. Therefore, Proposition 7.2 is indeed a surprising result proving that linear network codes are not optimal in general.

VIII. Conclusion

Characterisation of the set of zero-error or vanishing-error achievable rate-capacity tuples for network coding is a fundamental problem in multiterminal information theory. In [4], it was proved that this characterisation problem is extremely difficult in general and is as hard as determining the set of all information inequalities. This goes some way toward explaining why the problem has so far been solved only for a few special cases.

The authors in [1] and [2] used entropy functions to implicitly characterise the set of achievable rate-capacity tuples for general networks. Although this characterisation is implicit, it offers insights about the structure of the set of achievable tuples. For example, knowing that the set of almost entropic functions $\bar{\Gamma}^*$ is not polyhedral, [4] proved that the set of achievable tuples also is not polyhedral in general.

This paper extended [1] and [2] in several aspects. First, we proved that when sources are colocated, the outer bound given in [2, Section 15.5] is indeed tight. In particular, we showed that the set of rate-capacity tuples achievable with vanishing error, and the set achievable with zero error are indeed the same. We also gave evidence to support our conjecture that the outer bound in [2, Section 15.5] remains tight even when sources are not colocated.

Secondly, we considered network coding problems subject to several practically-motivated constraints, such as linear coding, the restriction of some or all nodes to perform only routing, and security requirements. For these cases we characterised the set of zero-error and vanishing-error achievable rate-capacity tuples. Finally in Section VII, we proved that even for very simple network coding problems, such as the incremental multicast problem and the single source secure network coding problem with arbitrary wiretapping patterns, characterisation of achievable tuples is as hard as the characterisation problem for general unconstrained network coding. We also proved that linear codes are suboptimal for both the general incremental multicast problem and for the single source secure network coding problem.

Appendix A
Proof of Proposition 3.3

Consider the following combinatorial problem. Suppose that there are k boxes, t of which are nonempty. If we randomly select m distinct boxes, then the probability that all selected boxes are empty is upper bounded by

$$\Pr(\text{all } m \text{ boxes are empty}) \leq \left(1 - \frac{t}{k}\right)^m \tag{180}$$

Let $\kappa(c) = (1-c)^{1/c}$. Since $\lim_{c \to 0^+} \kappa(c) = \exp(-1)$, there exists $0 < \delta < 1$ such that $\kappa(c) < \delta$ for all $0 < c \leq 1$. Hence, (180) can be relaxed as

$$\Pr(\text{all } m \text{ boxes are empty}) \leq \delta^{tm/k}. \tag{181}$$

Let (U, V) be a pair of quasi-uniform random variables. As V is uniform over its support, $|\mathsf{SP}(V)| = 2^{H(V)}$. Let

$$m = H\left(UV\right)^2 \frac{2^{H(V)}}{2^{H(V|U)}}.$$

Partition the set $\mathsf{SP}(V)$ randomly into

$$2^{H(V)}/m = \frac{2^{H(V|U)}}{H\left(UV\right)^2}$$

subsets, each of which is of size m. These disjoint subsets will be denoted by $\Xi(b)$ where

$$b \in \mathcal{A}_V \triangleq \{1, \ldots, 2^{H(V|U)}/H\left(UV\right)^2\}.$$

For any $u \in \mathsf{SP}(U)$ and $b \in \mathcal{A}_V$, let $\mathbb{E}(u, b)$ be the event that

$$\{(u, i) : i \in \Xi(b)\} \cap \mathsf{SP}(U, V) \neq \emptyset.$$

In other words, the event is equivalent to the existence of an element $i \in \Xi(b)$ such that $(u, i) \in \mathsf{SP}(U, V)$.

In the following, we will prove that the probability of $\mathbb{E}(u, b)$ is "arbitrarily close to one asymptotically" for all $u \in \mathsf{SP}(U)$ and $b \in \mathcal{A}_V$.

For any $u \in \mathsf{SP}(U)$, it is easy to see that

$$|\{v : (u, v) \in \mathsf{SP}(U, V)\}| = 2^{H(V|U)}. \tag{182}$$

By setting $k = 2^{H(V)}$ and $t = 2^{H(V|U)}$, (181) implies that

$$\Pr(\mathbb{E}(u, b)) \geq 1 - \delta^{H(UV)^2} \tag{183}$$

and hence via the union bound, the probability that the event $\mathbb{E}(u, b)$ occurs for all $u \in \mathsf{SP}(U)$ and $b \in \mathcal{A}_V$ is at least

$$1 - 2^{H(U)+H(V)} \delta^{H(UV)^2}.$$

This probability approaches to 0 as $H\left(UV\right)$ goes to infinity. Consequently, if the entropy $H\left(U\right)$ (and hence also $H\left(UV\right)$) is sufficiently large, there exists a way to partition $\mathsf{SP}(V)$ such that for any $u \in \mathsf{SP}(U)$ and $b \in \mathcal{A}_V$, there exists at least one $v \in \Xi(b)$ such that $(u, v) \in \mathsf{SP}(U, V)$.

Assume without loss of generality that $\mathcal{S} = \{1, \dots, |\mathcal{S}|\}$. Repeating the same argument, we can recursively prove that for any set of quasi-uniform random variables $\{U_i, i \in \mathcal{S}\}$ and $H(U_1)$ sufficiently large, there exists at least a way to partition $\mathsf{SP}(U_s)$ into

$$2^{H(U_i|U_1,\dots,U_{i-1})}/H(U_1,\dots,U_i)^2$$

subsets $\Xi_s(b_s)$ where

$$b_s \in \mathcal{A}_s \triangleq \{1, \dots, 2^{H(U_s|U_1,\dots,U_{s-1})}/H(U_1,\dots,U_s)^2\}$$

such that for any $(u_1, \dots, u_{s-1}) \in \mathsf{SP}(U_1, \dots, U_{s-1})$ and $b_s \in \mathcal{A}_s$, there exists at least one $u_s \in \Xi_s(b_s)$ such that $(u_1, \dots, u_s) \in \mathsf{SP}(U_1, \dots, U_s)$. Hence, the proposition is proved.

Appendix B
Proof of Theorem 7.1

We first prove the first claim. Suppose

$$(\lambda^\dagger, \omega^\dagger) \triangleq T^\dagger(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}_{\mathsf{P}^\dagger}[h^\dagger])$$

for some

$$h^\dagger \in \bar{\Gamma}^*(\mathsf{P}^\dagger) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}^\dagger) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}^\dagger) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P}^\dagger).$$

Then by definition,

$$\lambda^\dagger(i) \le h^\dagger(i), \quad i = 1, 2 \tag{184}$$

$$\omega^\dagger(a_s) \ge h^\dagger(a_s), \quad \forall s \in \mathcal{S} \tag{185}$$

$$\omega^\dagger(b_s) \ge h^\dagger(b_s), \quad \forall s \in \mathcal{S} \tag{186}$$

$$\omega^\dagger(c_s) \ge h^\dagger(c_s), \quad \forall s \in \mathcal{S} \tag{187}$$

$$\omega^\dagger(e) \ge h^\dagger(e), \quad \forall e \in \mathcal{E}. \tag{188}$$

Consequently, by (171)–(174) and (185)–(187),

$$2\sum_{s\in\mathcal{S}} \lambda(s) = \sum_{s\in\mathcal{S}} (\omega^\dagger(a_s) + \omega^\dagger(b_s)) \ge \sum_{s\in\mathcal{S}} (h^\dagger(a_s) + h^\dagger(b_s)) \overset{(i)}{\ge} h^\dagger(a_s, b_s, s \in \mathcal{S}) \tag{189}$$

where (i) follows from the fact that $h^\dagger \in \bar{\Gamma}^*(\mathsf{P}^\dagger)$ (and hence is polymatroidal). Similarly, we can also prove that

$$2\sum_{s\in\mathcal{S}} \lambda(s) = \sum_{s\in\mathcal{S}} (\omega^\dagger(a_s) + \omega^\dagger(c_s)) \ge \sum_{s\in\mathcal{S}} (h^\dagger(a_s) + h^\dagger(c_s)) \ge h^\dagger(a_s, c_s, s \in \mathcal{S}). \tag{190}$$

Let g be the "projection" of $h^\dagger$ on $\mathcal{H}[\mathcal{S} \cup \mathcal{E}]$ such that for any $\alpha \subseteq \mathcal{E}$ and $\beta \subseteq \mathcal{S}$,

$$g(\alpha, \beta) \triangleq h^\dagger(\alpha, a_i, i \in \beta). \tag{191}$$

In the following, we will prove that

$$(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}_{\mathsf{P}}[g])$$

and

$$g \in \bar{\Gamma}^*(\mathsf{P}) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P}).$$

First, $h^\dagger \in \bar{\Gamma}^*(\mathsf{P}^\dagger)$. Hence, its projection g is also in $\bar{\Gamma}^*(\mathsf{P})$. Second, the network $\mathsf{G}^\dagger$ contains G as a subnetwork and $O(s) \subseteq \mathsf{head}(a_s)$. In other words, if a node u has access to the source s in the network coding problem P, then u also has access to what is being transmitted along the link a_s in $\mathsf{P}^\dagger$. The link a_s in $\mathsf{G}^\dagger$ is thus like an imaginary source link in G. It can then be verified directly that $g \in \mathcal{C}_{\mathsf{T}}(\mathsf{P})$.

Now, we will prove that $g \in \mathcal{C}_{\mathsf{I}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P})$. As

$$h^\dagger \in \bar{\Gamma}^*(\mathsf{P}^\dagger) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P}^\dagger) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}^\dagger),$$

and that the set of links $\{a_s, b_s, s \in \mathcal{S}\}$ separates the source node ϕ from the sink node η in $\mathsf{G}^\dagger$,

$$h^\dagger(1', 2' | a_s, b_s, s \in \mathcal{S}) = h^\dagger(a_s, b_s, s \in \mathcal{S} | 1', 2') = 0. \tag{192}$$

Consequently,

$$\begin{aligned} h^\dagger(a_s, b_s, s \in \mathcal{S}) &\geq h^\dagger(1', 2') && (193)\\ &\stackrel{(i)}{=} h^\dagger(1') + h^\dagger(2') && (194)\\ &\stackrel{(ii)}{\geq} \lambda^\dagger(1) + \lambda^\dagger(2) && (195)\\ &= 2\sum_{s \in \mathcal{S}} \lambda(s) && (196) \end{aligned}$$

where (i) follows from the fact that $h^\dagger \in \bar{\Gamma}^*(\mathsf{P}^\dagger) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}^\dagger)$ and (ii) follows from (184).

Similarly, the set of links $\{a_s, c_s, s \in \mathcal{S}\}$ separates the source node ϕ from the sink node η^* in $\mathsf{G}^\dagger$. Hence,

$$h^\dagger(a_s, c_s, s \in \mathcal{S}) \geq 2\sum_{s \in \mathcal{S}} \lambda(s). \tag{197}$$

Therefore, all the inequalities in (189) and (190) are in fact equalities. In particular,

$$h^\dagger(a_s) = h^\dagger(b_s) = h^\dagger(c_s) = \lambda(s), \quad \forall s \in \mathcal{S} \tag{198}$$

and

$$h^\dagger(a_s, b_s, s \in \mathcal{S}) = \sum_{s \in \mathcal{S}} (h^\dagger(a_s) + h^\dagger(b_s)), \tag{199}$$

$$h^\dagger(a_s, c_s, s \in \mathcal{S}) = \sum_{s \in \mathcal{S}} (h^\dagger(a_s) + h^\dagger(c_s)). \tag{200}$$

By (199), $h^\dagger(a_s, s \in \mathcal{S}) = \sum_{s \in \mathcal{S}} h^\dagger(a_s)$. Hence,

$$g(\mathcal{S}) = \sum_{s \in \mathcal{S}} g(s)$$

and $g \in \mathcal{C}_{\mathsf{I}}(\mathsf{P})$. Furthermore, as

$$g(s) = h^\dagger(a_s) = \lambda(s), \quad \forall s \in \mathcal{S} \tag{201}$$

$$g(e) = h^\dagger(e) \leq \omega(e), \quad \forall e \in \mathcal{E}, \tag{202}$$

we prove that

$$(\lambda, \omega) \in \mathsf{CL}(\mathsf{proj}_{\mathsf{P}}[g]).$$

Now, it remains to show that $g \in \mathcal{C}_{\mathsf{D}}(\mathsf{P})$. First, consider any $s \in \mathcal{S}$ and $u \in D(s)$. By (199)–(200),

$$h^{\dagger}\left(b_{\mathcal{S} \setminus s} \wedge a_{\mathcal{S}}, b_s\right) = 0.$$

As $h^{\dagger} \in \bar{\Gamma}^{*}(\mathsf{P}^{\dagger}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}^{\dagger})$, $h^{\dagger}(c_s \mid a_s, b_s) = h^{\dagger}(\mathsf{in}(u) \mid a_{\mathcal{S}}) = 0$. Hence,

$$h^{\dagger}(b_{\mathcal{S} \setminus s} \wedge a_{\mathcal{S}}, b_s, c_s, \mathsf{in}(u)) = 0. \tag{203}$$

Together with the decoding constraint (for the receiver $\tau_{s,u}$)

$$h^{\dagger}(a_s \mid b_{\mathcal{S} \setminus s}, c_s, \mathsf{in}(u)) = 0,$$

we can prove that

$$h^{\dagger}(a_s \mid c_s, \mathsf{in}(u)) = 0.$$

Finally, using (200) and that $h^{\dagger}(\mathsf{in}(u) \mid a_{\mathcal{S}}) = 0$, we have $h^{\dagger}(a_s \mid \mathsf{in}(u)) = 0$. Thus,

$$g(a_s \mid \mathsf{in}(u)) = h^{\dagger}(a_s \mid \mathsf{in}(u)) = 0$$

and $g \in \mathcal{C}_{\mathsf{D}}(\mathsf{P})$. The first claim is proved.

To prove the second claim, suppose (λ, ω) is 0-achievable with respect to P. By definition, there exists a sequence of zero-error network codes

$$\{Y_f^n, f \in \mathcal{S} \cup \mathcal{E}\}$$

for P, and a sequence of positive constants c_n such that

$$\lim_{n \to \infty} c_n H(Y_e^n) \leq \lim_{n \to \infty} c_n H|\mathsf{SP}(Y_e^n)| \leq \omega(e) \tag{204}$$

$$\lim_{n \to \infty} c_n H(Y_s^n) = \lim_{n \to \infty} c_n H|\mathsf{SP}(Y_s^n)| \geq \lambda(s). \tag{205}$$

Assume without loss of generality that

$$\mathsf{SP}(Y_s^n) = \{0, \ldots, |\mathsf{SP}(Y_s^n)| - 1\}.$$

For each n, define a new set of random variables

$$\{U_f^n, f \in \mathcal{S}^{\dagger} \cup \mathcal{E}^{\dagger}\}$$

such that for any $s \in \mathcal{S}$ and $u \in D(s)$,

1) $U_{a_s}^n \triangleq Y_s^n$;
2) $U_e^n \triangleq Y_e^n$;
3) $\{U_{b_s}^n, s \in \mathcal{S}\}$ is a set of mutually independent random variables such that each of which is uniformly distributed over $\{0, \ldots, |\mathsf{SP}(Y_s^n)| - 1\}$ and

$$H\left(U_{a_{\mathcal{S}}}^n, U_{\mathcal{E}}^n, U_{b_{\mathcal{S}}}^n\right) = \sum_{f \in \mathcal{S}} U_{b_f}^n + H\left(U_{a_{\mathcal{S}}}^n, U_{\mathcal{E}}^n\right);$$

4) $U_{c_s}^n \triangleq U_{a_s}^n + U_{b_s}^n \mod |Y_s^n|$;
5) $U_{d_{s,u}}^n \triangleq U_{a_s}^n$;
6) $U_{1'}^n \triangleq (U_{b_f}^n, f \in \mathcal{S})$;
7) $U_{2'}^n \triangleq (U_{a_f}^n, f \in \mathcal{S})$.

It can then be proved directly that $\{U_f^n, f \in \mathcal{S}^\dagger \cup \mathcal{E}^\dagger\}$ is a sequence of zero-error network codes for the network coding problem $\mathsf{P}^\dagger$. Consequently, $T(\lambda, \omega)$ is 0-achievable with respect to $\mathsf{P}^\dagger$. The theorem is proved.

Appendix C
Proof of Theorem 7.3

We first prove the first claim. Let

$$(\lambda^\ddagger, \omega^\ddagger) \triangleq T^\ddagger(\lambda, \omega) \in \mathsf{CL}(\mathsf{proj}_{\mathsf{P}^\ddagger}(h^\ddagger)) \tag{206}$$

for some

$$h^\ddagger \in \bar{\Gamma}^*(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{I}(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{T}(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{D}(\mathsf{P}^\ddagger) \cap \mathcal{C}_\mathsf{S}(\mathsf{P}^\ddagger). \tag{207}$$

By (175)-(206), for all $e \in \mathcal{E}, s \in \mathcal{S}$ and $u \in D(s)$,

$$\sum_{i \in \mathcal{S}} \lambda(i) = \lambda^\ddagger(1^{'}) \leq h^\ddagger(1^{'}) \tag{208}$$

$$\lambda(s) = \omega^\ddagger(a_s) \geq h^\ddagger(a_s) \tag{209}$$

$$\lambda(s) = \omega^\ddagger(b_s) \geq h^\ddagger(b_s) \tag{210}$$

$$\lambda(s) = \omega^\ddagger(c_s) \geq h^\ddagger(c_s) \tag{211}$$

$$\lambda(s) = \omega^\ddagger(d_{s,u}) \geq h^\ddagger(d_{s,u}) \tag{212}$$

$$\lambda(s) = \omega^\ddagger(w_{s,u}) \geq h^\ddagger(w_{s,u}) \tag{213}$$

$$\sum_{i \in \mathcal{S}, i \neq s} \lambda(i) = \omega^\ddagger(e_s) \geq h^\ddagger(e_s) \tag{214}$$

$$\omega(e) = \omega^\ddagger(e) \geq h^\ddagger(e). \tag{215}$$

Using (208)–(215), we have

$$h^\ddagger(1^{'}) \geq \sum_{s \in \mathcal{S}} \lambda(s) = \sum_{s \in \mathcal{S}} \omega^\ddagger(a_s) \geq \sum_{s \in \mathcal{S}} h^\ddagger(a_s) \overset{(i)}{\geq} h^\ddagger(a_\mathcal{S}). \tag{216}$$

where (i) follows from that $h^\ddagger \in \bar{\Gamma}^*(\mathsf{P}^\ddagger)$ and hence is a polymatroid. Similarly,

$$h^\ddagger(1^{'}) \geq \sum_{s \in \mathcal{S}} \lambda(s) = \sum_{s \in \mathcal{S}} \omega^\ddagger(b_s) \geq \sum_{s \in \mathcal{S}} h^\ddagger(b_s) \geq h^\ddagger(b_\mathcal{S}). \tag{217}$$

Recall that $h^\ddagger$ is a rank function in the space $\mathcal{H}[\mathcal{S}^\ddagger \cup \mathcal{E}^\ddagger \cup \mathcal{V}^\ddagger]$. Let g be its "projection" on $\mathcal{H}[\mathcal{S} \cup \mathcal{E}]$ such that for any $\alpha \subseteq \mathcal{E}$ and $\beta \subseteq \mathcal{S}$,

$$g(\alpha, \beta) \triangleq h^\ddagger(\alpha, a_i, i \in \beta \mid \mathcal{V}^\ddagger \setminus \{\phi\}). \tag{218}$$

In the following, we will prove that

$$(\lambda, \omega) \in \mathsf{CL}(\mathrm{proj}_{\mathsf{P}}[g])$$

and

$$g \in \bar{\Gamma}^*(\mathsf{P}) \cap \mathcal{C}_{\mathsf{I}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}) \cap \mathcal{C}_{\mathsf{D}}(\mathsf{P}).$$

First, as $h^{\ddagger} \in \bar{\Gamma}^*(\mathsf{P}^{\ddagger})$, it is obvious that

$$g \in \bar{\Gamma}^*(\mathsf{P}).$$

Second, notice that the network $\mathsf{G}^{\ddagger}$ contains G as a subnetwork. Therefore, by $h^{\ddagger} \in \bar{\Gamma}^*(\mathsf{P}^{\ddagger}) \cap \mathcal{C}_{\mathsf{T}}(\mathsf{P}^{\ddagger})$,

$$g \in \mathcal{C}_{\mathsf{T}}(\mathsf{P}).$$

Now, as $h^{\ddagger} \in \mathcal{C}_{\mathsf{I}}(\mathsf{P}^{\ddagger}) \cap \mathcal{C}_{\mathsf{S}}(\mathsf{P}^{\ddagger})$, we have

$$h^{\ddagger}(1', \mathcal{V}^{\ddagger}) = h^{\ddagger}(1') + \sum_{u \in \mathcal{V}^{\ddagger}} h^{\ddagger}(u) \tag{219}$$

$$h^{\ddagger}\left(1' \wedge a_{\mathcal{S}}\right) = 0 \tag{220}$$

$$h^{\ddagger}\left(1' \wedge b_{\mathcal{S}}\right) = 0. \tag{221}$$

By (219), we can deduce that

$$h^{\ddagger}\left(1', \phi \wedge \mathcal{V}^{\ddagger} \setminus \{\phi\}\right) = 0. \tag{222}$$

Due to the topology constraint (for the links $\{a_s, b_s, s \in \mathcal{S}\}$),

$$h^{\ddagger}(a_{\mathcal{S}}, b_{\mathcal{S}} \mid 1', \phi) = 0. \tag{223}$$

Hence,

$$h^{\ddagger}\left(a_{\mathcal{S}}, b_{\mathcal{S}}, 1' \wedge \mathcal{V}^{\ddagger} \setminus \{\phi\}\right) = 0 \tag{224}$$

On the other hand, the decoding constraint $h^{\ddagger} \in \mathcal{C}_{\mathsf{D}}(\mathsf{P})$ (for the sink node η), we have

$$h^{\ddagger}(1' \mid a_{\mathcal{S}}, b_{\mathcal{S}}) = 0. \tag{225}$$

Together with (220)–(221), (and with the fact that $h^{\ddagger}$ is a polymatroid) (225) implies that

$$h^{\ddagger}(a_{\mathcal{S}}) \geq h^{\ddagger}(1') \tag{226}$$

$$h^{\ddagger}(b_{\mathcal{S}}) \geq h^{\ddagger}(1'). \tag{227}$$

By the upper bounds on $h^{\ddagger}(a_{\mathcal{S}})$ and $h^{\ddagger}(b_{\mathcal{S}})$ in (216)–(217), we can in fact prove that

$$h^{\ddagger}\left(a_{\mathcal{S}} \wedge b_{\mathcal{S}}\right) = 0 \tag{228}$$

$$h^{\ddagger}(a_{\mathcal{S}}, b_{\mathcal{S}}) = \sum_{s \in \mathcal{S}} (h^{\ddagger}(a_s) + h^{\ddagger}(b_s)). \tag{229}$$

$$h^{\ddagger}(a_s, s \in \mathcal{S}) = \sum_{s \in \mathcal{S}} h^{\dagger}(a_s) \tag{230}$$

$$h^{\ddagger}(a_s) = h^{\ddagger}(b_s) = \lambda(s), \quad \forall s \in \mathcal{S}. \tag{231}$$

Now, for any $s \in \mathcal{S}$,

$$
\begin{aligned}
g(s) &= h^{\ddagger}(a_s \mid \mathcal{V}^{\ddagger} \setminus \{\phi\}) \\
&\stackrel{(i)}{=} h^{\ddagger}(a_s) \\
&= \lambda(s)
\end{aligned}
$$

where (i) follows from (224). Also, for any $e \in \mathcal{E}$,

$$
\begin{aligned}
g(e) &= h^{\ddagger}(e \mid \mathcal{V}^{\ddagger} \setminus \{\phi\}) \\
&\leq h^{\ddagger}(e) \\
&\leq \omega(e).
\end{aligned}
$$

Therefore, $(\lambda, \omega) \in \mathsf{CL}(\text{proj}_{\mathsf{P}}[g])$.

By definition,

$$
g(\mathcal{S}) = h^{\ddagger}(a_{\mathcal{S}} \mid \mathcal{V}^{\ddagger} \setminus \{\phi\}) \stackrel{(i)}{=} h^{\ddagger}(a_{\mathcal{S}}) = \sum_{s \in \mathcal{S}} h^{\ddagger}(a_s) \geq \sum_{s \in \mathcal{S}} h^{\ddagger}(a_s \mid \mathcal{V}^{\ddagger} \setminus \{\phi\}) = \sum_{s \in \mathcal{S}} g(s) \geq g(\mathcal{S}) \tag{232}
$$

where (i) is due to (224). Thus, $g \in \mathcal{C}_{\mathsf{I}}(\mathsf{P})$.

Our last step is to prove that $g \in \mathcal{C}_{\mathsf{D}}(\mathsf{P})$. As

$$
h^{\ddagger}(c_s) + h^{\ddagger}(e_s) \leq \sum_{s \in \mathcal{S}} \lambda_s = h^{\ddagger}(1^{'}),
$$

the decoding constraint $h^{\ddagger}(1^{'} \mid c_s, e_s) = 0$ (for the receiver ψ_s) implies that

$$
h^{\ddagger}(c_s \mid 1^{'}) = 0 \tag{233}
$$

and

$$
h^{\ddagger}(c_s) = \lambda(s) \tag{234}
$$

By (220)–(221) and (229)

$$
h^{\ddagger}(c_s \wedge a_s) = h^{\ddagger}(c_s \wedge b_s) = h^{\ddagger}(a_s \wedge b_s) = 0 \tag{235}
$$

On the other hand, by (219), $h^{\ddagger}\left(\gamma_s \wedge 1^{'}, \phi\right) = 0$. By (223) and (233), we have $h^{\ddagger}\left(\gamma_s \wedge a_s, b_s, c_s\right) = 0$. Together with one of the topology constraint (for the link c_s)

$$
h^{\ddagger}(c_s \mid a_s, b_s, \gamma_s) = 0, \tag{236}
$$

we have

$$
h^{\ddagger}(c_s \mid a_s, b_s) = 0. \tag{237}
$$

By (234) and (231), we can prove that

$$
h^{\ddagger}(c_s \mid a_s, b_s) = h^{\ddagger}(a_s \mid c_s, b_s) = h^{\ddagger}(b_s \mid a_s, c_s) = 0. \tag{238}
$$

Similarly, focusing on the receiver $\tau_{s,u}$, the decoding constraint $h\left(1^{'} \mid w_{s,u}, e_s\right) = 0$ and that $h^{\ddagger}(w_{s,u}) \leq \lambda(s)$ imply that

$$h^{\ddagger}(w_{s,u} \mid 1^{'}) = 0 \tag{239}$$

and

$$h^{\ddagger}(w_{s,u}) = \lambda(s). \tag{240}$$

By (221), $h^{\ddagger}(w_{s,u} \wedge b_s) = 0$. On the other hand, by (224) and (220)

$$h^{\ddagger}(1^{'} \wedge a_{\mathcal{S}}, \mathcal{V}^{\ddagger} \setminus \{\phi\}) = h^{\ddagger}(1^{'}, a_{\mathcal{S}} \wedge \mathcal{V}^{\ddagger} \setminus \{\phi\}) = 0. \tag{241}$$

Furthermore, by the topology constraint

$$h^{\ddagger}(d_{s,u} \mid a_{\mathcal{S}}, \mathcal{V}^{\ddagger} \setminus \{\phi\}) = 0 \tag{242}$$

Therefore, together with (239), we have

$$h^{\ddagger}\left(1^{'}, w_{s,u} \wedge d_{s,u}, a_{\mathcal{S}}\right) = 0. \tag{243}$$

Similarly, by (224) and (228),

$$h^{\ddagger}(b_{\mathcal{S}} \wedge a_{\mathcal{S}}, \mathcal{V}^{\ddagger} \setminus \{\phi\}) = h^{\ddagger}(a_{\mathcal{S}}, b_{\mathcal{S}} \wedge \mathcal{V}^{\ddagger} \setminus \{\phi\}) = 0 \tag{244}$$

and hence

$$h^{\ddagger}\left(b_{\mathcal{S}} \wedge d_{s,u}, a_{\mathcal{S}}\right) = 0. \tag{245}$$

Consequently, we have

$$h^{\ddagger}(w_{s,u} \wedge b_s) = h^{\ddagger}(w_{s,u} \wedge d_{s,u}) = h^{\ddagger}(d_{s,u} \wedge b_s) = 0. \tag{246}$$

On the other hand, by the topology constraint,

$$h^{\ddagger}(w_{s,u} \mid b_s, d_{s,u}, \theta_{s,u}) = 0. \tag{247}$$

Again, by (224), $h^{\ddagger}\left(1^{'}, b_s, d_{s,u} \wedge \theta_{s,u}\right) = 0$. Then, by (247), we can prove that

$$h^{\ddagger}(w_{s,u} \mid b_s, d_{s,u}) = 0. \tag{248}$$

Using (240), (246) and (248) and (212) and (231), we can prove that

$$h^{\ddagger}(w_{s,u} \mid b_s, d_{s,u}) = h^{\ddagger}(d_{s,u} \mid b_s, w_{s,u}) = h^{\ddagger}(b_s \mid w_{s,u} d_{s,u}) = 0 \tag{249}$$

and

$$h^{\ddagger}(d_{s,u}) = \lambda(s). \tag{250}$$

Finally, notice that

$$h^{\ddagger}\left(b_s \wedge a_s d_{s,u}\right) \overset{(i)}{=} 0 \tag{251}$$

$$h^{\ddagger}\left(w_{s,u}, c_s \wedge a_s d_{s,u}\right) \overset{(ii)}{=} 0 \tag{252}$$

$$h^{\ddagger}\left(b_s \wedge w_s c_s\right) \overset{(iii)}{=} 0 \tag{253}$$

where (i), (ii) and (iii) follow respectively (245), (243) and (221). By (238) and (249), we have

$$h^{\ddagger}(b_s \mid w_{s,u} c_s, a_s d_{s,u}) = 0 \tag{254}$$

$$h^{\ddagger}(a_s d_{s,u} \mid b_s, w_{s,u} c_s,) = 0. \tag{255}$$

Together with (251)–(253), we can prove that

$$h^{\ddagger}(a_s d_{s,u}) = h^{\ddagger}(b_s) = \lambda(s) = h^{\ddagger}(d_{s,u}).$$

Thus, $h^{\ddagger}(a_s \mid d_{s,u}) = 0$ and

$$g(a_s \mid \mathsf{in}(u)) = h^{\ddagger}(a_s \mid \mathsf{in}(u), \mathcal{V}^{\ddagger} \setminus \{\phi\}) = 0.$$

Thus, $g \in \mathcal{C}_{\mathsf{D}}(\mathsf{P})$ and the first claim is proved.

We will now prove the second claim. The idea of the proof is similar to that in the incremental multicast scenario. Suppose (λ, ω) is 0-achievable with respect to P. By definition, there exists a sequence of zero-error network codes

$$\{Y_f^n, f \in \mathcal{S} \cup \mathcal{E}\}$$

for P, and a sequence of positive constants c_n such that

$$\lim_{n\to\infty} c_n H(Y_e^n) \leq \lim_{n\to\infty} c_n H|\mathsf{SP}(Y_e^n)| \leq \omega(e) \tag{256}$$

$$\lim_{n\to\infty} c_n H(Y_s^n) = \lim_{n\to\infty} c_n H|\mathsf{SP}(Y_s^n)| \geq \lambda(s). \tag{257}$$

Assume without loss of generality that $\mathsf{SP}(Y_s^n)$ (i.e., the support of Y_s^n) is equal to $\{0, \ldots, |\mathsf{SP}(Y_s^n)| - 1\}$. For each n, construct the following set of random variables

$$\{U_f, f \in \mathcal{S}^{\ddagger} \cup \mathcal{E}^{\ddagger} \cup \mathcal{V}^{\ddagger}\}$$

such that for all $e \in \mathcal{E}, s \in \mathcal{S}$ and $u \in D(s)$,

1) $U_{a_s}^n = U_{d_{s,u}}^n = Y_s^n$;
2) $U_e^n = Y_e^n$;
3) $U_{b_s}^n$ is uniformly distributed over $\mathsf{SP}(Y_s^n)$ for all $s \in \mathcal{S}$ and that
$$H\left(U_{a_{\mathcal{S}}}^n, U_{\mathcal{E}}^n, U_{b_{\mathcal{S}}}^n\right) = \sum_{f\in\mathcal{S}} U_{b_f}^n + H\left(U_{a_{\mathcal{S}}}^n, U_{\mathcal{E}}^n\right);$$
4) $U_{w_{s,u}}^n = U_{c_s}^n = Y_{a_s}^n + Y_{b_s}^n \mod |\mathsf{SP}(Y_s^n)|$;
5) $U_{e_s}^n \triangleq (Y_{c_i}^n, i \in \mathcal{S} \setminus s)$
6) $U_{1'}^n = (Y_{c_s}^n, s \in \mathcal{S})$;

7) $U_v^n = 1$ (i.e., $U_v = 1$ is a deterministic random variable) for all $v \in \mathcal{V}^\ddagger$;

Again, it can be verified directly that

$$\{U_f^n, f \in \mathcal{S}^\ddagger \cup \mathcal{E}^\ddagger \cup \mathcal{V}^\ddagger\}$$

is a strongly secure zero-error network codes for $\mathsf{P}^\ddagger$. Consequently, $T(\lambda, \omega)$ is 0-achievable with respect to $P^\ddagger$, subject to strong secrecy constraint.

References

[1] X. Yan, R. W. Yeung, and Z. Zhang, "The capacity region for multi-source multi-sink network coding," in *IEEE Int. Symp. Inform. Theory*, 2007.

[2] R. Yeung, *A First Course in Information Theory*. Kluwer Academic/Plenum Publisher, 2002.

[3] F. Matus, "Infinitely many information inequalities," in *Proceedings ISIT 2007, June 2007, (Nice, France)*, 2007, pp. 41–44.

[4] T. H. Chan and A. Grant, "Dualities between entropy functions and network codes," *IEEE Trans. Inform. Theory*, vol. 54, no. Oct, pp. 4470–4487, 2008.

[5] S.-Y. R. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[6] N. Cai and R. Yeung, "Secure network coding," in *IEEE Int. Symp. Inform. Theory*, 2002.

[7] N. Harvey, R. Kleinberg, and A. Lehman, "On the capacity of information networks," *IEEE Trans. Inform. Theory*, vol. 52, pp. 2345–2364, June 2006.

[8] G. Kramer and S. A. Savari, "Edge-cut bounds on network coding rates," *J. Netw. Syst. Manage.*, vol. 14, pp. 49–67, March 2006.

[9] S. Thakor, A. Grant, and T. Chan, "Network coding capacity: A functional dependence bound," in *2009 Proc. on IEEE International Symposium on Information Theory*, July 2009, pp. 263–267.

[10] R. Yeung, "A framework for linear information inequalities," *IEEE Trans. Inform. Theory*, vol. 43, no. 6, pp. 1924–1934, Nov. 1997.

[11] T. H. Chan, "Recent progresses in characterising information inequalities," *Entropy Journal*, vol. 13, pp. 379–401, 2011.

[12] Z. Zhang and R. W. Yeung, "On the characterization of entropy function via information inequalities," *IEEE Trans. Inform. Theory*, vol. 44, pp. pp. 1440–1452, 1998.

[13] L. Song, R. Yeung, and N. Cai, "Zero-error network coding for acyclic networks," *IEEE Trans. Inform. Theory*, vol. 49, no. 12, pp. 3129–3139, Dec. 2003.

[14] T. H. Chan and R. W. Yeung, "On a relation between information inequalities and group theory," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1992–1995, 2002.

[15] J. Oxley, *Matroid Theory*. Oxford University Press, 1992.

[16] Y. Wu, P. A. Chou, and K. Jain, "A comparison of network coding and tree packing," in *IEEE Int. Symp. Inform. Theory*, Chicago, USA, 2004.

[17] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[18] A. Shamir, "How to share a secret," *Comm. ACM*, vol. 22, pp. 612–613, 1979.

[19] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: John Wiley, 1991.

[20] N. Cai and R. W. Yeung, "Secure network coding," *to appear in IEEE Trans. Inf. Theory*, 2011.

[21] ——, "A security condition for multi-source linear network coding," in *IEEE Int. Symp. Inform. Theory*, 2007.